CAPÍTULO 5

CATEGORIZACIÓN DE DOCUMENTOS CON *BLINDLIGHT*

La categorización de documentos es el proceso por el cual se asocian una o más categorías a textos escritos en un lenguaje natural basándose tan sólo en su contenido. Aunque es posible construir de manera "manual" un categorizador, las técnicas estadísticas y, por tanto, automáticas son actualmente las preferidas puesto que no sólo ofrecen un rendimiento muy adecuado sino que resulta mucho más sencillo seleccionar un conjunto de ejemplos para entrenar un algoritmo que elaborar reglas manualmente. Así pues, esta tarea entra dentro del campo del aprendizaje automático y es posible aplicarle una gran variedad de técnicas disponibles. En este capítulo se presentará la categorización de documentos en tanto que problema de aprendizaje, se mostrarán algunas de las posibles aplicaciones de la categorización automática de texto libre, se revisarán las distintas técnicas aplicadas al problema y, por último, se describirá la forma de utilizar blindLight como categorizador y se presentarán los resultados obtenidos con esta nueva técnica.

1 Categorización automática de documentos

Según el diccionario de la *RAE* (2001) la categorización es la acción y efecto de organizar o clasificar mediante categorías, entendidas éstas como un elemento de clasificación. La definición no es muy clara pero no parece prudente adentrarse en terreno filosófico y sí en cambio proponer una definición de categorización útil en el campo del tratamiento de información. Así, podría definirse la categorización como la acción que realiza un agente al etiquetar *ítems* con una o más categorías de un conjunto predefinido basándose en las características de los mismos.

La categorización así entendida se viene utilizando desde hace siglos. Aristóteles[1] o Linneo, por ejemplo, establecieron categorías de plantas y animales. Posteriormente, se

[1] Aristóteles fue quien acuño el término "categoría" a partir del término *katigoro* (κατηγορώ), "acusar", "predicar" de algo o alguien.

desarrollaron diversos métodos para organizar los recursos almacenados en bibliotecas como el de la Biblioteca del Congreso de EE.UU. (*Library of Congress Classification*, *LCC*) o el Sistema Decimal Universal (*Universal Decimal Classification*, *UDC*). Más reciente es el sistema de categorización de la *ACM* (*ACM Computing Classification System*) para clasificar artículos sobre informática y otros para categorizar recursos web como por ejemplo la jerarquías desarrolladas por *Yahoo!* (Steinberg 1996) o al amparo del *Open Directory Project* (*ODP*).

La mayor parte de estos sistemas estructuran las categorías en jerarquías lo cual es en ocasiones un inconveniente[1] (véase Fig. 73) y según algunos autores simplemente obsoleto (Bates 2002). No obstante, puesto que, hasta donde sabe el autor, no es posible la obtención automática de categorías para un sistema alternativo como la clasificación por facetas[2] y dado que la mayor parte de técnicas de categorización automática no suelen aprovechar tal estructura jerárquica[3] este capítulo se centrará en la categorización automática de documentos en categorías "planas", sin ningún tipo de relación jerárquica entre las mismas.

H. Information Systems
H.3 INFORMATION STORAGE AND RETRIEVAL
H.3.3 Information Search and Retrieval
Clustering
...
I. Computing Methodologies
I.5 PATTERN RECOGNITION
I.5.3 **Clustering**

"The cascade of interactions in the digital library interface"
(Marcia J. Bates 2002)

H. Information Systems
H.3 INFORMATION STORAGE AND RETRIEVAL
H.3.3 Information Search and Retrieval
H.3.5 On-line Information Services
H.3.7 Digital Libraries
H.5 INFORMATION INTERFACES AND PRESENTATION (I.7)
H.5.2 User Interfaces (D.2.2, H.1.2, I.3.6)

Fig. 73 Fragmento del sistema de clasificación de la *ACM* y categorización de un artículo.

A la izquierda se muestra un fragmento del sistema de clasificación de la ACM, nótese cómo una de las categorías (*clustering*) aparece como subcategoría de dos categorías distintas. A la derecha se muestra la clasificación de un artículo sobre interfaces de usuario de bibliotecas digitales; además de requerirse cuatro categorías para etiquetar dicho trabajo varias de las categorías están cruzadas (hacia I.7 Procesamiento de Documentos y Texto, D.2.2 Herramientas y Técnicas de Diseño, H.1.2 Sistemas Persona/Máquina e I.3.6 Metodología y Técnicas).

Las categorizaciones antes mencionadas fueron elaboradas por expertos humanos; sin embargo, la categorización automática de documentos no sólo es posible sino que constituye una herramienta muy útil para enfrentarse a la consabida sobrecarga de información.

Una aplicación inmediata es la de asignar temas a los documentos o *topic tagging*. Por ejemplo, Maarek y Ben Shaul (1996) emplearon esta técnica para organizar listas de enlaces favoritos (*bookmarks*), Hearst y Karadi (1997) para etiquetar grupos de documentos médicos extrayendo las categorías más frecuentes en cada conjunto y Attardi, Gullì y Sebastiani (1999) para etiquetar documentos web.

---

[1] Steinberg (1996, p. 3) relata una anécdota sobre la categorización de un sitio web en el directorio *Yahoo!* El sitio web de la *Messianic Jewish Alliance of America* (Alianza Judía Mesiánica de América) fue inicialmente adscrito a la categoría "Judaismo". Este hecho provocó quejas de ciertas organizaciones judías puesto que consideran a los judíos mesiánicos unos herejes ya que creen que Jesús es el mesías. Ante las protestas el sitio fue trasladado a la categoría "Cristiandad" causando entonces las quejas de la citada organización puesto que no son cristianos. Finalmente la página fue asignada a una categoría propia, "Judaismo Mesiánico", que actualmente contiene 220 enlaces.

[2] Sistematizada por S.R. Ranganathan (Chan 1994, p. 390).

[3] No obstante, se ha tratado de explotar la estructura jerárquica para mejorar el rendimiento de los categorizadores de documentos (Koller y Sahami 1997), (McCallum *et al.* 1998) o (Weigend, Wiener y Pedersen 1999)

No obstante, en estos trabajos no puede hablarse propiamente de categorización automática. En el primer y tercer caso no hay categorías predefinidas ya que las etiquetas se extraen automáticamente, mientras que en el segundo, aunque se utilizaban las categorías *MeSH*, no se empleaban para categorizar documentos sino para etiquetar *clusters*.

Chekuri *et al.* (1997), en cambio, emplearon las categorías de la taxonomía de *Yahoo!* para etiquetar sitios web y Li *et al.* (1999) al igual que Maarek y Ben Shaul desarrollaron un sistema para organizar *bookmarks* pero, al contrario que ellos, no utilizaron etiquetas automáticas sino las definidas en otras taxonomías (como por ejemplo la empleada por la Biblioteca del Congreso de EE.UU).

También se han aplicado técnicas de categorización automática para clasificar correo electrónico en categorías establecidas por el usuario (Cohen 1996), filtrar correo no deseado o *spam* (Sahami *et al.* 1998) o determinar si un documento es o no relevante para un usuario dado (Schütze, Hull y Pedersen 1995).

2 La categorización como un problema de aprendizaje automático

Las aplicaciones anteriormente descritas se enmarcan dentro de los problemas de aprendizaje supervisado que tiene como objetivo la obtención de una función a partir de datos de entrenamiento. En el caso de la categorización automática de documentos los datos de entrada serán vectores construidos a partir de los documentos y los valores de salida serán discretos (la categoría[1] a la que se asigna cada documento).

Ni que decir tiene que existen muy diversas técnicas para obtener dicha función y que se han aplicado muchas de ellas a la categorización de texto natural. Fabrizio Sebastiani (2002) publicó un magnífico estudio sobre el tema revisando aspectos fundamentales como la forma de representar los documentos, las distintas técnicas para construir categorizadores así como la manera de evaluar dichos categorizadores. No obstante, al igual que en capítulos anteriores, se describirán brevemente algunas de las técnicas más comunes.

Una forma sencilla de categorizar documentos consiste en examinar el texto en busca de algún término que permita discriminarlo. El correo electrónico constituye un magnífico ejemplo puesto que existen dos categorías principales: mensajes deseados y no deseados. Así, aquellos que incluyan términos como `congratulations`, `won`, `free`, `unlimited` o `mortgage` pueden desecharse con relativa confianza. Un usuario podría especificar reglas en su cliente de correo para eliminar este tipo de mensajes. Sin embargo, poco después tendría que añadir reglas para variaciones "tipográficas" de los mismos términos como `mortage` (*sic*) o `un1imited`. Por tanto, lo ideal sería emplear técnicas que, a partir de una serie de ejemplos, generasen las reglas apropiadas. Los árboles y reglas de decisión o, mejor dicho, los métodos para inducirlos son tales técnicas.

Un **árbol de decisión** constituye una agrupación jerárquica de reglas que permiten obtener un único valor objetivo a partir de un conjunto de datos de origen (véase Fig. 74). Las **reglas de decisión** son similares pero dan lugar a categorizadores más compactos. Por su propia naturaleza este tipo de técnicas son más intuitivas para el usuario final que podría, en caso necesario, añadir reglas propias. No obstante, aunque esto resulta una ventaja obvia los términos no suelen ponderarse sino que deben manejarse de forma binaria. Aun así, hay toda una serie de trabajos que han aplicado árboles –Lewis y Catlett (1994), Apté, Damerau y Weiss (1998) o Weiss *et al.* (1999)– o reglas de decisión –Apté, Damerau y Weiss (1994),

[1] Como se verá más adelante, en muchos problemas no resulta viable limitarse a una única categoría por cada documento y, así, a cada *ítem* se asociarán una o más categorías o etiquetas.

Cohen (1996), Moulinier y Ganascia (1996) o Li y Yamanishi (1999)– a la categorización automática de documentos y se han utilizado árboles de decisión como base de otros métodos de categorización como el *boosting* (que se verá más adelante).

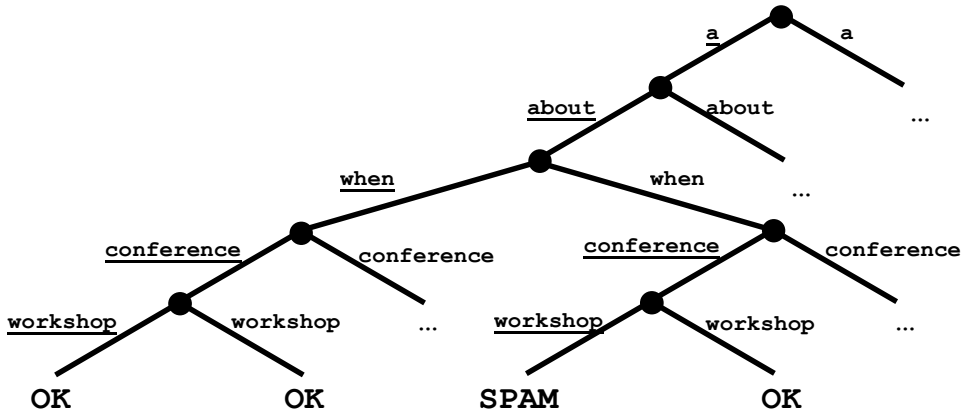

Fig. 74 Un árbol de decisión incompleto y muy sencillo para separar el *spam* recibido por un investigador hispanohablante.

Las ramas están etiquetadas con términos que pueden aparecer en los documentos. Los términos subrayados indican la ausencia del término y las hojas representan las distintas categorías.

Como se ha dicho, los árboles y reglas de decisión emplean una representación binaria de los documentos; sin embargo, es posible construir un categorizador de documentos basado en el modelo vectorial clásico de una forma muy sencilla. En la fase de "entrenamiento" simplemente se construye el vector para cada documento de la colección y se le asocia su categoría. En la fase de categorización se emplea el documento a clasificar como consulta y se obtienen k resultados, la clase más frecuente entre esos k documentos resultantes es la categoría a la que pertenece el nuevo documento.

Debido al trabajo prácticamente nulo que se realiza durante el entrenamiento este método es calificado como "perezoso" (Chakrabarti 2003, p. 134), por otro lado, requiere almacenar toda la información de entrenamiento y resulta computacionalmente costoso en el momento de la categorización. No obstante, puede refinarse y ofrecer resultados superiores a los obtenidos con árboles o reglas de decisión (Cohen y Hirsh 1998) (Han, Karypis y Kumar 1999) o próximos a los de técnicas más eficaces como *SVM* (Yang y Liu 1999). Una de las modificaciones más relevantes en esta técnica es el cambio del método de ponderación puesto que la técnica *tf*idf* aplicada sobre la colección de documentos de entrenamiento no es muy efectiva (Yang y Chute 1994, citado por Han *et al.* 1999) o (Chakrabarti 2003, p. 135).

Otro tipo de categorizadores muy populares son los denominados ***naïve Bayes*** (Minsky y Papert 1969, citado en Elkan 1997, p.3). Se trata de categorizadores probabilísticos basados en el teorema de Bayes y que reciben el apelativo *naïve* (simple) al suponer, de manera deliberada e irreal[1], una total independencia entre los valores que toman los distintos términos en cada clase. En la fase de entrenamiento este categorizador calcula

[1] El hecho de que los términos no son independientes entre sí resulta especialmente claro si se toma como ejemplo la poblada categoría de correo no solicitado; términos como `v1agra`, `anonymously`, `prescription`, `online` u `order` muestran una clara dependencia.

la probabilidad de aparición de cada término (normalmente palabras o *stems*) condicionada a cada categoría. En la fase de categorización se debe calcular la probabilidad de cada categoría condicionada a la frecuencia de aparición de los términos en el documento seleccionando la más probable.

A pesar de su simplicidad los categorizadores *naïve* Bayes muestran un buen rendimiento (Domingos y Pazzani 1997, p. 105-106) y, según Chakrabarti (2003, p. 175-176), tal vez su sencillez, su facilidad de implementación y la rapidez con que se adaptan a cambios en las colecciones de documentos explican su popularidad frente a otros algoritmos más efectivos. A pesar de todo, su aplicación a categorización de documentos es relativamente reciente: por ejemplo, Larkey y Croft (1996) los emplearon, individualmente y combinados con otros métodos, para asignar códigos *ICD*[1] a diagnósticos médicos, Koller y Sahami (1997) y Chakrabarti *et al.* (1998c) para categorizar documentos dentro de taxonomías y Sahami *et al.* (1998) para filtrar correo no deseado.

Otra técnica de aprendizaje automático que también se ha utilizado para categorizar documentos son las redes neuronales. Una **red neuronal** (McCulloch y Pitts 1943) (Rosenblatt 1958) es un sistema que conecta un conjunto de elementos de proceso muy simples en una serie de capas. La capa de entrada contiene tantos elementos como variables definan los ejemplares del problema y la capa de salida un elemento por cada categoría a detectar.

Schütze, Hull y Pedersen (1995), Wiener, Pedersen y Weigend (1995), Weigend, Wiener y Pedersen (1999), Ng, Goh y Low (1997) o Ruiz y Srinivasan (1997 y 1999) han aplicado con éxito redes neuronales a problemas de categorización de documentos. Yang y Liu (1999) mostraron que los resultados obtenidos con esta técnica son inferiores a los alcanzados con *k*-vecinos o *SVM* (que se verá más adelante) mientras que Lam y Lee (1999) analizaron en qué medida la reducción del número de términos puede mejorar el rendimiento de la red neuronal.

Ya se mencionaron en el capítulo anterior los mapas de Kohonen o mapas auto-organizativos (*SOM*) así como su relación con las redes neuronales. Roussinov y Chen (1998) y Ontrup y Ritter (2001) la han utilizado para categorizar documentos. Tan sólo los últimos ofrecen una comparativa con otra técnica, en este caso *k*-vecinos, afirmando que los resultados obtenidos por *SOM* se aproximan a los de dicho método.

Además del método de los *k*-vecinos hay otra técnica de categorización que no surgió del campo del aprendizaje automático sino del área de recuperación de información; se trata del **algoritmo de Rocchio** (1971) para expandir consultas por realimentación (*relevance feedback*). La idea es sencilla: (1) dada una consulta, un sistema de recuperación de información proporciona al usuario un conjunto de documentos, (2) el usuario selecciona los que considera relevantes y (3) se "enriquece" la consulta original calculando la diferencia entre los documentos relevantes (POS_i, véase la ecuación) y los no relevantes (NEG_i). Así, una categoría c_i estaría representada por un vector de pesos w_{ki} calculados según la siguiente fórmula en la que w_{kj} es el peso del término t_k en el documento d_j.

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|}$$

[1] *ICD – International Statistical Classification of Diseases and Related Health Problems* (Clasificación Estadística Internacional de Enfermedades y otros Problemas de la Salud) es un catálogo publicado por la *OMS* que describe de manera detallada enfermedades y heridas a las que asigna un código de hasta 5 caracteres.

Según Sebastiani (2002) fue Hull (1994) el primero en adaptar esta técnica, sin embargo, ese trabajo trata sobre un caso especial de categorización, la separación de una colección en conjuntos de documentos relevantes y no relevantes a partir de unos pocos ejemplares. A juicio del autor[1] sería más acertado atribuir el mérito de la adaptación a Ittner, Lewis y Ahn (1995) que emplearon Rocchio para asignar textos obtenidos a partir de imágenes de baja calidad a distintas categorías. Además de estos, otros investigadores han empleado o mejorado el algoritmo de Rocchio en este campo, por ejemplo Joachims (1997), Singhal, Mitra y Buckley (1997) o Schapire, Singer y Singhal (1998).

Spam-1	Powerful enlargement.
Spam-2	Find out about cialis. Viagra's big brother.
Spam-3	Viagra: save more buying more stylus sad.
Spam-4	Get viagra anonymously! Fast shipping.
Spam-5	Viagra, vallium, cialis.
Ham-1	Call for participation: constraints in discourse.
Ham-2	Call for participation and studentship applications for CLIMA VI.
Ham-3	Fast SVM training on very large data sets.
Ham-4	Job opening for nationals of EU enlargement countries.
Ham-5	Call for papers ICCBSS 2006.

Fig. 75 Una colección de 10 documentos y 2 categorías: *spam* (correo no deseado) y *ham* (correo deseado).

Para aclarar el funcionamiento del método de Rocchio se describirá un pequeño ejemplo de categorización de correo electrónico. En Fig. 75 se muestra la colección que se empleará para el "entrenamiento" y que consta de 10 breves documentos que pertenecen las categorías de correo deseado (*ham*) y no deseado (*spam*).

Descontando las palabras vacías se obtienen 34 términos distintos[2] para los que se calcula su valor *idf*. Posteriormente, se determina para cada documento su representación vectorial otorgando a cada término del documento su peso *tf*idf*. No obstante, puesto que no hay términos repetidos en ninguno de los textos, este peso resulta idéntico al valor *idf* original (véase Fig. 76).

Como se puede apreciar en la ecuación anterior el método de Rocchio es parametrizable mediante los valores β y γ que, básicamente, permiten controlar qué tipo de ejemplos (positivos o negativos) influyen más a la hora de construir la correspondiente categoría. En Fig. 77 y Fig. 79 se muestra el resultado de emplear $\beta=\gamma=1$ (ejemplos positivos y negativos tienen la misma influencia) y en Fig. 80 para $\beta=0{,}5$ y $\gamma=1$ (los ejemplos negativos tienen más influencia que los positivos).

Con independencia de los valores otorgados a ambos parámetros es necesario calcular el peso w_{ki} para cada término k de cada categoría i de acuerdo a la ecuación anterior (véase Fig. 77 y Fig. 80). Una vez calculado el módulo del vector de cada categoría ya ha finalizado el "entrenamiento" del categorizador.

Para llevar a cabo la categorización de un nuevo documento bastaría con comparar el vector correspondiente con los de las distintas categorías. Aquella categoría más próxima sería aquella a la que habría que asignar al documento. A fin de mostrar este funcionamiento se utilizarán los documentos que aparecen en Fig. 78.

[1] Juicio compartido por Cohen y Singer (1999, p. 155).

[2] Los términos se representan con el subíndice k en la ecuación.

Término	IDF	Spam-1	Spam-2	Spam-3	Spam-4	Spam-5	Ham-1	Ham-2	Ham-3	Ham-4	Ham-5
2006	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**
anonymously	1,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00
applications	1,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00
big	1,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
brother	1,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
buying	1,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00
cialis	0,50	0,00	**0,50**	0,00	0,00	**0,50**	0,00	0,00	0,00	0,00	0,00
clima	1,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00
constraints	1,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00
countries	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00
data	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
discourse	1,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00
enlargement	0,50	**0,50**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**0,50**	0,00
eu	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00
fast	0,50	0,00	0,00	0,00	**0,50**	0,00	0,00	0,00	0,50	0,00	0,00
iccbss	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**
job	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00
large	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
nationals	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00
opening	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00
papers	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**
participation	0,50	0,00	0,00	0,00	0,00	0,00	**0,50**	**0,50**	0,00	0,00	0,00
powerful	1,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sad	1,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00
save	1,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sets	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
shipping	1,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00
studentship	1,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00
stylus	1,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00
svm	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
training	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
vallium	1,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00
vi	1,00	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00	0,00
viagra	0,25	0,00	**0,25**	**0,25**	**0,25**	**0,25**	0,00	0,00	0,00	0,00	0,00

Fig. 76 La colección anterior representada mediante el modelo vectorial.

Se han eliminado las palabras vacías y se ha simplificado el valor de *idf* como el inverso del número de documentos que contienen un término dado.

Término	Spam ✓	Spam ✖	WkSpam (β=γ=1)	Término	Ham ✓	Ham ✖	WkHam (β=γ=1)
2006	0,00	0,20	-0,20	2006	0,20	0,00	0,20
anonymously	0,20	0,00	0,20	anonymously	0,00	0,20	-0,20
applications	0,00	0,20	-0,20	applications	0,20	0,00	0,20
big	0,20	0,00	0,20	big	0,00	0,20	-0,20
brother	0,20	0,00	0,20	brother	0,00	0,20	-0,20
buying	0,20	0,00	0,20	buying	0,00	0,20	-0,20
cialis	0,20	0,00	0,20	cialis	0,00	0,20	-0,20
clima	0,00	0,20	-0,20	clima	0,20	0,00	0,20
constraints	0,00	0,20	-0,20	constraints	0,20	0,00	0,20
countries	0,00	0,20	-0,20	countries	0,20	0,00	0,20
data	0,00	0,20	-0,20	data	0,20	0,00	0,20
discourse	0,00	0,20	-0,20	discourse	0,20	0,00	0,20
enlargement	0,10	0,10	0,00	enlargement	0,10	0,10	0,00
eu	0,00	0,20	-0,20	eu	0,20	0,00	0,20
fast	0,10	0,10	0,00	fast	0,10	0,10	0,00
iccbss	0,00	0,20	-0,20	iccbss	0,20	0,00	0,20
job	0,00	0,20	-0,20	job	0,20	0,00	0,20
large	0,00	0,20	-0,20	large	0,20	0,00	0,20
nationals	0,00	0,20	-0,20	nationals	0,20	0,00	0,20
opening	0,00	0,20	-0,20	opening	0,20	0,00	0,20
papers	0,00	0,20	-0,20	papers	0,20	0,00	0,20
participation	0,00	0,20	-0,20	participation	0,20	0,00	0,20
powerful	0,20	0,00	0,20	powerful	0,00	0,20	-0,20
sad	0,20	0,00	0,20	sad	0,00	0,20	-0,20
save	0,20	0,00	0,20	save	0,00	0,20	-0,20
sets	0,00	0,20	-0,20	sets	0,20	0,00	0,20
shipping	0,20	0,00	0,20	shipping	0,00	0,20	-0,20
studentship	0,00	0,20	-0,20	studentship	0,20	0,00	0,20
stylus	0,20	0,00	0,20	stylus	0,00	0,20	-0,20
svm	0,00	0,20	-0,20	svm	0,20	0,00	0,20
training	0,00	0,20	-0,20	training	0,20	0,00	0,20
vallium	0,20	0,00	0,20	vallium	0,00	0,20	-0,20
vi	0,00	0,20	-0,20	vi	0,20	0,00	0,20
viagra	0,20	0,00	0,20	viagra	0,00	0,20	-0,20
		|spam|	**1,13**			**|ham|**	**1,13**

Fig. 77 Cálculo de los vectores para las categorías *spam* y *ham* (β=γ=1).

Empleando los mismos valores de β y γ (véase la ecuación en página 111) se "valoran" en la misma medida los ejemplos positivos y negativos para calcular el vector que representa a cada categoría.

La comparación entre los vectores de documento y categorías puede realizarse con cualquier medida pero en general se emplea la función del coseno (véase Fig. 79 y Fig. 80).

Los métodos descritos hasta el momento (p.ej. árboles y reglas de decisión, categorizadores bayesianos o Rocchio) emplean un único agente que "aprende" a partir de un único conjunto de ejemplos. Sin embargo, es posible emplear varios categorizadores y constituir lo que se denomina un **"comité"** de tal manera que cada miembro del mismo emita un voto para cada documento a categorizar. Una técnica distinta de los comités pero que también requiere la participación de varios categorizadores es la denominada ***boosting***[1]. La aproximación intuitiva es la siguiente: (1) un categorizador "aprende" sobre una parte del conjunto de entrenamiento y es probado sobre el resto del conjunto[2]; (2) aquellos documentos del conjunto de entrenamiento que clasifique mal, junto con algunos otros de su subconjunto de entrenamiento original, se utilizan para entrenar otro categorizador que, de este modo, "aprende" casos más "difíciles"; (3) el esquema se repite *n* veces.

spamTest Cheapest viagra, cialis delivered anonymously.
hamTest EU must defer enlargement if French vote no.

Término	spamTest	hamTest
2006	0,00	0,00
anonymously	**1,00**	0,00
applications	0,00	0,00
big	0,00	0,00
brother	0,00	0,00
buying	0,00	0,00
cialis	**0,50**	0,00
clima	0,00	0,00
constraints	0,00	0,00
countries	0,00	0,00
data	0,00	0,00
discourse	0,00	0,00
enlargement	0,00	**0,50**
eu	0,00	**1,00**
fast	0,00	0,00
iccbss	0,00	0,00
job	0,00	0,00
large	0,00	0,00
nationals	0,00	0,00
opening	0,00	0,00
papers	0,00	0,00
participation	0,00	0,00
powerful	0,00	0,00
sad	0,00	0,00
save	0,00	0,00
sets	0,00	0,00
shipping	0,00	0,00
studentship	0,00	0,00
stylus	0,00	0,00
svm	0,00	0,00
training	0,00	0,00
vallium	0,00	0,00
vi	0,00	0,00
viagra	**0,25**	0,00

Fig. 78 Un par de documentos de prueba y su traducción al modelo vectorial.

Naturalmente, es necesario garantizar que semejante método produce sistematicamente un aprendizaje efectivo. Esa garantía fue obtenida por Schapire a partir del trabajo de Valiant y Kearns. Valiant (1984) introdujo el modelo de aprendizaje ***PAC*** (*Probably Aproximately Correct*) en el cual el categorizador recibe ejemplos de una clase tomados al azar y debe producir una regla que permita discriminar nuevos ejemplares. Un categorizador "eficiente" encuentra una regla correcta, con una alta probabilidad, para todos los ejemplares excepto para una fracción establecida de manera arbitraria. Kearns (1988) introdujo el concepto de categorizador "débil", aquel cuya regla de decisión es sólo ligeramente mejor que una decisión al azar, y planteó la posibilidad de alcanzar un categorizador eficiente partiendo de categorizadores débiles (la denominada ***boosting hypothesis***).

[1] Literalmente "aumento", "promoción", "elevación", "empuje".

[2] Recuérdese en las colecciones empleadas para el entrenamiento y prueba de métodos de categorización los documentos deben estar "etiquetados" con su correspondiente categoría (o categorías).

WkSpamTest·WkSpam	WkSpamTest·WkHam	WkHamTest·WkSpam	WkHamTest·WkHam
0,00	0,00	0,00	0,00
0,20	-0,20	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,10	-0,10	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	-0,20	0,20
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00
0,05	-0,05	0,00	0,00

spamTest·spam	spamTest·ham	hamTest·spam	hamTest·ham
0,35	-0,35	-0,20	0,20

cosDis (spamTest, spam) β=γ=1	cosDis (spamTest, ham) β=γ=1	cosDis (hamTest, spam) β=γ=1	cosDis (hamTest, ham) β=γ=1
0,27	-0,27	-0,16	**0,16**

Fig. 79 Cálculo de la función del coseno entre los documentos de prueba y las categorías (β=γ=1).

Si se admiten vectores con pesos negativos (como es el caso) la función del coseno toma valores entre -1 (180º) y +1 (0º). Así, a mayor valor mayor similitud y en este caso la categoría más próxima será aquella a la que se asigne el documento. En el ejemplo el documento *spamTest* ha sido etiquetado correctamente como *spam* y *hamTest* como correo deseado.

Término	Spam ✓	Spam ✖	WkSpam (β=0.5, γ=1)
2006	0,00	0,20	-0,20
anonymously	0,20	0,00	0,10
applications	0,00	0,20	-0,20
big	0,20	0,00	0,10
brother	0,20	0,00	0,10
buying	0,20	0,00	0,10
cialis	0,20	0,00	0,10
clima	0,00	0,20	-0,20
constraints	0,00	0,20	-0,20
countries	0,00	0,20	-0,20
data	0,00	0,20	-0,20
discourse	0,00	0,20	-0,20
enlargement	0,10	0,10	-0,05
eu	0,00	0,20	-0,20
fast	0,10	0,10	-0,05
iccbss	0,00	0,20	-0,20
job	0,00	0,20	-0,20
large	0,00	0,20	-0,20
nationals	0,00	0,20	-0,20
opening	0,00	0,20	-0,20
papers	0,00	0,20	-0,20
participation	0,00	0,20	-0,20
powerful	0,20	0,00	0,10
sad	0,20	0,00	0,10
save	0,20	0,00	0,10
sets	0,00	0,20	-0,20
shipping	0,20	0,00	0,10
studentship	0,00	0,20	-0,20
stylus	0,20	0,00	0,10
svm	0,00	0,20	-0,20
training	0,00	0,20	-0,20
vallium	0,20	0,00	0,10
vi	0,00	0,20	-0,20
viagra	0,20	0,00	0,10
		|spam|	**0,96**

cosDis (spamTest, spam) β=0.5, γ=1	cosDis (spamTest, ham) β=0.5, γ=1
0,22	-0,44

Término	Ham ✓	Ham ✖	WkHam (β=0.5, γ=1)
2006	0,20	0,00	0,10
anonymously	0,00	0,20	-0,20
applications	0,20	0,00	0,10
big	0,00	0,20	-0,20
brother	0,00	0,20	-0,20
buying	0,00	0,20	-0,20
cialis	0,00	0,20	-0,20
clima	0,20	0,00	0,10
constraints	0,20	0,00	0,10
countries	0,20	0,00	0,10
data	0,20	0,00	0,10
discourse	0,20	0,00	0,10
enlargement	0,10	0,10	-0,05
eu	0,20	0,00	0,10
fast	0,10	0,10	-0,05
iccbss	0,20	0,00	0,10
job	0,20	0,00	0,10
large	0,20	0,00	0,10
nationals	0,20	0,00	0,10
opening	0,20	0,00	0,10
papers	0,20	0,00	0,10
participation	0,20	0,00	0,10
powerful	0,00	0,20	-0,20
sad	0,00	0,20	-0,20
save	0,00	0,20	-0,20
sets	0,20	0,00	0,10
shipping	0,00	0,20	-0,20
studentship	0,20	0,00	0,10
stylus	0,00	0,20	-0,20
svm	0,20	0,00	0,10
training	0,20	0,00	0,10
vallium	0,00	0,20	-0,20
vi	0,20	0,00	0,10
viagra	0,00	0,20	-0,20
		|ham|	**0,83**

cosDis (hamTest, spam) β=0.5, γ=1	cosDis (hamTest, ham) β=0.5, γ=1
-0,28	**0,09**

Fig. 80 Cálculo de los vectores para las categorías *spam* y *ham* y de la similitud entre documentos de prueba y categorías (β=0.5 y γ=1).

Los valores β y γ utilizados en este caso dan mayor importancia a los ejemplos negativos que a los positivos. Nuevamente la categorización de los documentos de prueba es correcta.

Schapire (1990) demostró la equivalentecia entre ambos tipos de aprendizaje y que, efectivamente, resultaba factible construir un categorizador eficiente (en el marco del aprendizaje *PAC*) partiendo de categorizadores débiles. El método de *boosting*, por tanto, emplea un algoritmo de categorización cualquiera y construye de manera secuencial una serie de categorizadores que tratan de mejorar los peores resultados obtenidos por el categorizador anterior. El propio Schapire es el responsable de la mayor parte de las aplicaciones del *boosting* a la categorización de documentos, por ejemplo, Schapire *et al.* (1998) o Schapire y Singer (2000).

Por último se describirá la técnica que ofrece los mejores resultados (Joachims 1998) (Dumais *et al.* 1998) (Kwok 1998) (Yang y Liu 1999), la conocida como ***Support Vector Machines*** o ***SVM***. Este método fue propuesto originalmente por Boser, Guyon y Vapnik (1992) y consiste en la transformación de los vectores de entrada, que definen una dimensión en la cual no son linealmente separables, a una dimensión superior que permita su separación mediante una única (hiper)superficie (véase Fig. 81).

La aplicación de la técnica *SVM* a categorización de texto se debe[1] a Joachims (1998) aunque tanto Dumais *et al.* (1998) como Kwok (1998) hicieron de forma casi simultánea la misma propuesta. A partir de ese momento otros investigadores han empleado

[1] Según Sebastiani (1999, p. 36).

SVM para categorizar documentos: Drucker, Wu y Vapnik (1999), Schohn y Cohn (2000), Tong y Koller (2000), Rennie y Rifkin (2001) o Jalam y Teytaud (2001).

Ya se ha mencionado que los categorizadores *naïve* Bayes son muy utilizados aun cuando el método *SVM* es superior. Tal vez la complejidad del método de entrenamiento, cuadrática, pudiese afectar negativamente a su popularización al precisar de conjuntos de entrenamiento relativamente pequeños y, en consecuencia, limitados. No obstante, Osuna, Freund y Girosi (1997), Kaufman (1998, citado en Platt 1999), Platt (1999), Cauwenberghs y Poggio (2000) o Collobert, Bengio y Bengio (2002) plantearon diversas formas de solucionar ese inconveniente acelerando la fase de entrenamiento y/o facilitando la manipulación de colecciones cambiantes. No obstante, la razón fundamental probablemente sea la "implementación intimidante" (Platt 1998) de las *SVMs* frente a métodos menos efectivos pero mucho más sencillos.

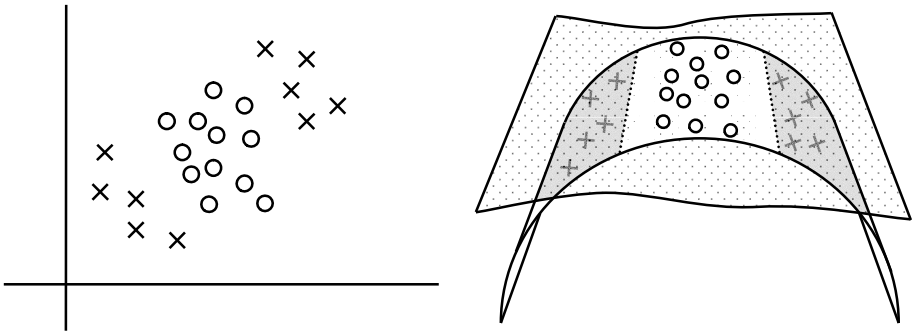

Fig. 81 Aproximación intuitiva al método de vectores soporte.

Los datos del problema original definen un espacio bidimensional donde no son linealmente separables. Transformandolos en un espacio tridimensional ya son separables mediante un plano.

3 Categorización de documentos con *blindLight*

Para categorizar una serie de documentos utilizando *blindLight* tanto los documentos como las categorías deben estar disponibles en la forma de vectores de *n*-gramas tal y como se describieron en los capítulos anteriores. Para obtener un vector de una categoría es posible emplear un único documento de muestra[1] aunque es más habitual realizar un entrenamiento sobre un conjunto de ejemplos. En ese caso el procedimiento es muy simple: (1) calcular un vector de *n*-gramas para cada documento de entrenamiento, (2) calcular el centroide de cada categoría, (3) calcular el centroide de todos los documentos de entrenamiento y (4) restar al centroide de cada categoría el centroide del conjunto de entrenamiento.

Una vez terminada la fase de aprendizaje se dispone de un único vector por categoría, vector susceptible de ser comparado con los vectores correspondientes a los documentos a categorizar. Aunque podría utilizarse la medida anteriormente descrita *PiRo* (véase página 66) ésta no proporciona resultados excesivamente satisfactorios. La razón es simple: dado que el número de *n*-gramas distintos aumenta[2] con la cantidad de documentos

[1] En el siguiente apartado se presentará un categorizador basado en *blindLight* que emplea un único documento para representar cada categoría.

[2] Crowder y Nicholas (1996) muestran cómo a medida que crece el número de documentos también aumenta el número de *n*-gramas distintos hasta alcanzar una meseta.

procesados los vectores de las distintas categorías y de los documentos a categorizar tienen tamaños y significatividades muy dispares. La diferencia en el tamaño es irrelevante pero una diferencia de significatividad notoria puede llevar a que las medidas Π y P tengan órdenes de magnitud diferentes y, por tanto, no tenga sentido combinarlas en una única medida de similitud.

Para evitar este problema se propone la siguiente versión "normalizada" de *PiRo*. Π es el cociente de la significatividad total de la intersección de documento y categoría entre la significatividad total del documento, P es el cociente de la significatividad total de la intersección entre la significatividad total de la categoría y *n* y *m* son, respectivamente, el número de *n*-gramas en los vectores categoría y documento. En todos los experimentos que se describen a lo largo del capítulo se ha utilizado esta medida con $\alpha=1-\alpha=0{,}5$[1].

$$PiRoNorm = \alpha \cdot \Pi + (1-\alpha) \cdot \frac{n}{m} \cdot \mathrm{P}$$

Por último, en la fase de categorización para cada documento incógnita se obtiene el vector de *n*-gramas correspondiente, y se calculan los valores de Π y P, y en consecuencia de *PiRoNorm*, para todas las categorías disponibles. En caso de tratarse de un problema de categorización en el que sólo deba asignarse una etiqueta al documento[2] se toma aquella categoría que obtenga para ese documento el valor máximo de *PiRoNorm*. En aquellos problemas en que puedan asignarse varias etiquetas a cada documento[3] tan sólo se proporciona una lista ordenada de todas las categorías y, por el momento, no se hace intento alguno por limitar el número de etiquetas asignadas.

4 Identificación automática del idioma a partir de un texto

Identificar el idioma en que está escrito un texto constituye un caso particular dentro de la categorización de documentos y es una tarea habitualmente requerida por buscadores web o en colecciones multi-idioma. Se han implementado múltiples sistemas para realizar este trabajo (Beesley 1988), (Cavnar y Trenkle 1994), (Dunning 1994), (Grefenstette 1995), (Sibun y Reynar 1996) o (Prager 1999) y todos han alcanzado una precisión casi perfecta. El único aspecto mejorable era el tiempo de ejecución y recientemente se ha presentado una nueva técnica (Poutsma 2002) que, aun cuando ofrece una precisión ligeramente peor que la mejor técnica disponible, es 85 veces más rápida que ésta.

En este sentido, la utilización de *blindLight* para identificar distintos idiomas sería una aportación poco significativa pues resulta muy difícil mejorar sustancialmente la precisión. No obstante, sí sería interesante determinar el rendimiento de esta nueva técnica en dos situaciones habituales al procesar información textual en Internet. A saber,

---

[1] El carácter *ad hoc* de esta medida no resulta totalmente satisfactorio. En la página 141 se muestran otras medidas de similitud y se discute la posibilidad de emplear programación genética para descubrir otras puesto que los valores Π y P son constantes para cada par de documentos comparados. No obstante, esto no invalida el hecho de que *blindLight* puede emplearse para llevar a cabo categorización automática, simplemente no se puede afirmar que esta medida sea la que ofrece una categorización óptima en todos los casos.

[2] Como por ejemplo en los experimentos descritos en los dos apartados siguientes: identificación de idioma y filtrado de correo no solicitado.

[3] Como en las colecciones `Reuters-21578` y `OHSUMED`.

documentos muy cortos (por ejemplo, el texto de consultas realizadas por los usuarios) y documentos con "ruido" (errores ortográficos, cabeceras de correo electrónico o *USENET*, etiquetas *HTML*, *Javascript*, etc).

A fin de comparar un identificador de idiomas basado en *blindLight* con otros métodos sería interesante disponer de alguna colección "estándar". Lamentablemente, la mayor parte de los autores elaboraron sus propias colecciones que no están disponibles; por su parte, Grefenstette y Sibun y Reynar sí emplearon una colección pública pero no libre[1].

Por suerte, algunos de los sistemas de identificación de idiomas están disponibles[2] *online*. Así, *TEXTCAT*[3] es una implementación del método de Cavnar y Trenkle capaz de identificar 70 idiomas distintos; *XEROX*[4] dispone de una herramienta aparentemente relacionada con los trabajos de Beesley y Grefenstette que soporta 47 idiomas y también existe una aplicación[5] de *Acquaintance* (Damashek 1995) mencionada en capítulos anteriores que distingue 66 idiomas y dialectos.

Según Poutsma (2002) el método que ofrece mayor precisión, incluso con muestras de unas decenas de caracteres, es el de Cavnar y Trenkle. En cuanto al sistema de *XEROX*, al ser de código cerrado (aunque se puede usar *online* de forma gratuita), no existe ninguna publicación reciente que describa la actual implementación ni analice su rendimiento de modo que el único "respaldo" a su eficiencia es el hecho de que se trata de un producto comercial. Así pues, al comparar *blindLight* con las tres herramientas citadas se estaría enfrentando a métodos cuya precisión ha sido sobradamente contrastada. Sin embargo, antes de describir los experimentos llevados a cabo es necesario comentar brevemente el modo en que las técnicas anteriores identifican los idiomas.

Tanto *TEXTCAT* como *Acquaintance* utilizan *n*-gramas de caracteres para construir el modelo de los lenguajes y representar las muestras de texto. La diferencia entre ambos radica en la manera en que se comparan muestra y modelo. Cavnar y Trenkle (1994) utilizan una medida denominada *out-of-place* ("fuera de lugar") que consiste básicamente en ordenar, basándose en su frecuencia de aparición, los *n*-gramas de muestra y modelo y determinar para cada *n*-grama de la muestra si ocupa el mismo puesto en el modelo o cuán desplazado se encuentra[6]. En el caso de *Acquaintance* se emplea el producto escalar.

Por lo que se refiere al sistema de *XEROX*, no es posible hacer ninguna afirmación categórica puesto que Grefenstette (1995) describe dos técnicas: una basada en el uso de trigramas de caracteres y otra basada en la utilización de palabras comunes[7]. No obstante, Langer (2001) describe un identificador híbrido (utilizado por el buscador *AllTheWeb*[8]) que primeramente trata de identificar el idioma empleando palabras comunes y sólo en caso de no obtener un resultado fiable recurre a los *n*-gramas. Por ello, resulta razonable suponer que el actual sistema de *XEROX* opere de manera similar empleando los dos métodos descritos por Grefenstette (1995) para identificar el idioma utilizado en un texto.

---

[1] *European Corpus Initiative CD-ROM*.

[2] Enero de 2005.

[3] `http://odur.let.rug.nl/~vannoord/TextCat/`

[4] `http://www.xrce.xerox.com/competencies/content-analysis/tools/guesser.en.html`

[5] `http://complingone.georgetown.edu/~langid/`

[6] Se trata de una medida similar en cierto modo al coeficiente de correlación de Spearman (pág. 75).

[7] Estas palabras comunes serían las "palabras vacías" mencionadas en otros capítulos.

[8] `http://www.alltheweb.com`

Es necesario señalar, además, que en ninguno de los casos existe información sobre la cantidad de texto que se ha empleado para construir el modelo de cada idioma. Es éste un dato especialmente relevante puesto que, según Dunning (1994), un sistema entrenado sobre 50 Kbytes de texto alcanza una precisión del 99,9% sobre muestras de 500 bytes frente al 97% de un sistema entrenado con sólo 5 Kbytes.

Una versión preliminar del identificador basado en *blindLight* se describe en (Gayo Avello *et al.* 2004b). Dicho sistema era capaz de identificar 14 idiomas[1] y se había "entrenado" a partir de documentos de unos 10 Kbytes, en todos los casos los tres primeros capítulos del libro del Génesis. Posteriormente se cambió el texto modelo, aunque el tamaño apenas cambió, y se incrementó el número de idiomas: se escogió la Declaración Universal de Derechos Humanos y los idiomas identificables pasaron a ser 37[2].

Por lo que respecta al método de categorización es muy simple. Para cada idioma se construye un vector de *n*-gramas a partir de los documentos modelo; al recibir una muestra de un texto desconocido se obtiene su correspondiente vector y éste es comparado mediante *PiRoNorm* (véase pág. 118) con los vectores anteriores. El idioma que muestre un mayor parecido será el asignado a la muestra.

Como se dijo antes se perseguían dos objetivos con los experimentos: por un lado, determinar el comportamiento de *blindLight* y otros identificadores frente a textos muy cortos y, por otro, comprobar la tolerancia al ruido de los distintos métodos.

Para llevar a cabo la primera prueba se tomaron los temas[3] utilizados en las campañas de 2003 y 2004 del *CLEF* (*Cross Language Evaluation Forum*) un ejemplo de los cuales aparece en Fig. 82. Los temas estaban disponibles en alemán, castellano, finés, francés e inglés para 2003 y 2004 además de en italiano y sueco para el último año. La campaña de 2003 ofreció 60 temas por idioma y la de 2004 50 temas. Para cada tema e idioma se elaboraron 7 documentos combinando los distintos elementos constituyentes[4] obteniendo de este modo 4.550 documentos que contenían desde una única palabra hasta cerca de 100.

Una vez elaborada esta colección cada uno de los sistemas identificadores fue aplicado sobre la misma anotando el idioma asignado a cada documento y comparando la identificación con el idioma en que estaba originalmente escrito[5]. Los resultados obtenidos se muestran en la Tabla 10 y la Tabla 11 y de ellos se deduce que la técnica propuesta por el autor es sustancialmente mejor que las de (Cavnar y Trenkle 1994) y (Damashek 1995) para textos de entre 1 y 5 palabras[6] pero no consigue superar al sistema de *XEROX* en esos

---

[1] Alemán, castellano, catalán, danés, faroés, finés, francés, holandés, inglés, italiano, noruego, portugués, sueco y vasco.

[2] A los anteriores se añadieron: asturiano, bretón, corso, croata, eslovaco, estonio, frisón, gaélico escocés, gaélico irlandés, galés, gallego, húngaro, islandés, latín, letón, lituano, maltés, occitano auvergnat, occitano languedoc, polaco, rumano, sardo y turco.

[3] En cada campaña *CLEF* se ofrece un conjunto de temas en varios idiomas para elaborar (de manera automática o manual) las consultas que se utilizarán para evaluar los sistemas *IR*. Estos temas tratan sobre un asunto específico y constan de tres partes: (1) un breve título, (2) una descripción más verbosa de la necesidad de información a satisfacer y (3) una serie de criterios que se emplearán para juzgar la relevancia de los documentos retornados por el sistema.

[4] Cada tema *CLEF* consta de título (T), descripción (D) y narración (N). Así, los documentos construidos para cada tema serían: T, D, N, TD, TN, DN y TDN.

[5] En algunos casos el documento no estaba escrito en el idioma de "consulta". Dos temas de *CLEF'04* tenían por títulos `Lady Diana` y `Christopher Reeve`.

[6] En realidad *blindLight* es sustancialmente superior a *TEXTCAT* para textos de hasta 20 palabras.

mismos documentos. Cabe pensar si el hecho de que este último emplee listas de palabras vacías (conocimiento lingüístico) como una de las "pistas" para identificar el idioma pueda influir en un rendimiento superior para los textos más cortos.

```
<top>
     <num>
          C154
     </num>
     <ES-title>
          Libertad de Expresión en Internet
     </ES-title>
     <ES-desc>
          Encontrar documentos en los que se hable sobre la
          censura y la libertad de expresión en Internet.
     </ES-desc>
     <ES-narr>
          Los documentos en los que se discutan asuntos
          como la pornografía o el racismo en Internet, sin
          mencionar el tema de la censura o libertad de
          expresión, no se considerarán relevantes.
     </ES-narr>
</top>
```

Fig. 82 Un tema para la campaña de 2003 del *CLEF* escrito en castellano.

Los temas del *CLEF* describen necesidades de información que deben ser resueltas por un sistema *IR* obteniendo documentos de una colección de artículos periodísticos. Cada tema incluye un título corto, una descripción (normalmente una oración que indica la consulta) y una descripción más larga de las cualidades deseables en los documentos relevantes y, ocasionalmente, características de documentos no relevantes.

Longitud del texto	*blindLight*	Xerox	TEXTCAT	*Acquaintance*
1 o 2 palabras	42,37%	54,62%	10,97%	31,22%
3 a 5 palabras	65,91%	79,24%	26,90%	56,85%
6 a 10 palabras	90,13%	94,57%	60,10%	84,55%
11 a 15 palabras	95,10%	98,04%	78,51%	94,54%
16 a 20 palabras	98,48%	99,67%	87,68%	97,71%
21 a 30 palabras	99,45%	99,80%	92,13%	98,99%
31 a 50 palabras	99,86%	100,00%	96,06%	99,92%
Más de 50 palabras	100,00%	100,00%	99,42%	99,88%

Tabla 10. Precisión (macropromediada, *macroaveraged*[1]) de los cuatro identificadores.

El identificador basado en *blindLight* es sustancialmente mejor que *Acquaintance* para textos de entre 1 y 20 palabras, apreciablemente mejor para textos de 21 a 30 y la diferencia es inapreciable para textos más largos. *blindLight* es sustancialmente mejor que *TEXTCAT* para textos de entre 1 y 5 palabras y las diferencias son inapreciables para textos más extensos. El identificador de *XEROX* supera a *blindLight* de manera sustancial al identificar textos de 1 a 5 palabras y ofrece resultados análogos con muestras mayores.

[1] David D. Lewis (1991) señala dos formas de obtener resultados promedio al evaluar un sistema de categorización a las que denomina *macro-* y *microaveraging*. Para un conjunto de D documentos y una serie de K categorías un categorizador toma $D \cdot K$ decisiones evaluables individualmente. A fin de obtener un valor promedio puede calcularse la media de las evaluaciones de las decisiones correspondientes a cada categoría (*macroaveraging*) o bien tomar las $D \cdot K$ decisiones en conjunto (*microaveraging*). La diferencia entre uno y otro método de evaluación es simple: en el caso de *microaveraging* tiene más influencia el resultado global (número total de categorizaciones correctas) frente a las diferencias entre categorías (puede haber diferencias notables entre los resultados obtenidos para cada categoría) mientras que en el caso de *macroaveraging* influyen más las diferencias entre categorías que los resultados tomados en su conjunto, es decir, se "premiaría" al categorizador que obtiene resultados similares en todas las categorías. Dependiendo de la aplicación debe decidirse qué tipo de resultados son preferibles y emplear un método u otro de promedio para evaluar las distintas técnicas. Si se considera que todos los idiomas son igualmente importantes a la hora de ser correctamente identificados entonces los identificadores deberían evaluarse empleando *macroaveraging*. En cambio, si se considera que esto no es así, bien por el número de hablantes o por la cantidad de documentos existentes, debería optarse por evaluar mediante *microaveraging*. Langer (2001) proporciona datos interesantes sobre la importancia relativa de distintos idiomas encontrados en el índice de *AllTheWeb*.

Longitud del texto	*blindLight*	Xerox	TEXTCAT	*Acquaintance*
1 o 2 palabras	49,03%	63,23%	37,42%	9,03%
3 a 5 palabras	64,61%	76,25%	55,34%	24,70%
6 a 10 palabras	91,72%	94,70%	88,41%	65,23%
11 a 15 palabras	96,53%	98,07%	95,18%	78,03%
16 a 20 palabras	98,84%	99,67%	97,19%	87,27%
21 a 30 palabras	99,56%	99,78%	98,90%	92,76%
31 a 50 palabras	99,82%	100,00%	99,91%	95,70%
Más de 50 palabras	100,00%	100,00%	99,81%	99,25%

Tabla 11. Precisión (micropromediada, *microaveraged*) de los cuatro identificadores.

Las diferencias entre identificadores al comparar los resultados micropromediados son similares a los encontradas al comparar los datos macropromediados.

Por lo que respecta al segundo experimento se empleó la colección `1500-5LNG`[1] elaborada por el propio autor (Gayo Avello *et al.* 2004b). Dicha colección consta de 1500 artículos publicados en los grupos `soc.culture.basque`, `catalan`, `french`, `galiza` y `german`, es decir, contiene documentos escritos, teóricamente, en vasco, catalán, francés y alemán. El objetivo era idéntico al de Cavnar y Trenkle (1994), obtener textos escritos presumiblemente en un idioma (por ejemplo, catalán en el caso de `soc.culture.catalan`) a fin de probar el identificador de manera sencilla.

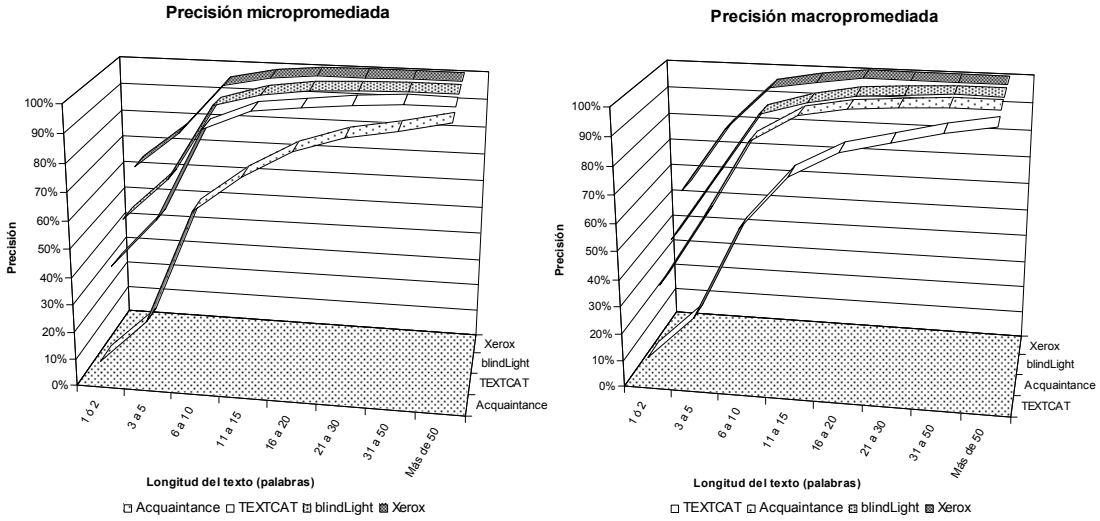

Fig. 83 Precisión de los identificadores en relación con la longitud del texto.

No obstante, estos grupos sufren de graves problemas de *spam* y publicación cruzada (*cross-posting*) por lo que los idiomas que aparecen en cada grupo son realmente diversos. Así, se emplean los siguientes idiomas: alemán, castellano, catalán, francés, gallego, inglés, italiano y vasco; mezclando en muchos artículos dos y, en ocasiones, tres idiomas. Por ello, y teniendo en cuenta que el sistema de *XEROX* no "conoce" el gallego, se eliminaron todos los artículos escritos en gallego (que no todos los artículos de `soc.culture.galiza`) además de aquellos en los que no había un predominio claro de un único idioma en el cuerpo del artículo. De este modo, la colección se redujo a 1358 documentos escritos en alemán, castellano, catalán, francés, inglés, italiano y vasco y que incluían las correspondientes cabeceras a modo de "ruido" (véase Fig. 84). Debido al escaso número de artículos en italiano y vasco (4 y 5, respectivamente) se evaluaron los distintos

[1] `http://www.di.uniovi.es/~dani/downloads/1500-5LNG.zip`

categorizadores con los restantes idiomas; los resultados obtenidos en este segundo experimento se muestran en Tabla 12 y Tabla 13.

```
From: unrien.dutout@nulle.part.fr (Unrien Dutout)
Newsgroups: soc.culture.french
Subject: C'est chouette ici...
Date: 4 Dec 2003 15:44:31 -0800
Organization: http://groups.google.com
Lines: 1
Message-ID: <67ec58c5.0312041544.218cc3d8@posting.google.com>
NNTP-Posting-Host: 82.66.227.13
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Enconding: 8bit
X-Trace: posting.google.com 1070581471 1210 127.0.0.1 (4 Dec 2003 23:44:31 GMT)
X-Complaints-To: groups-ab...@google.com
NNTP-Posting-Date: Thu, 4 Dec 2003 23:44:31 +0000 (UTC)

ça sent la déconfiture
```

Fig. 84 Un artículo escrito en francés con más del 90% de ruido.

Ruido en el texto	*blindLight*	Xerox	TEXTCAT	*Acquaintance*
0-5%	100,00%	99,24%	99,24%	98,47%
5-10%	100,00%	95,89%	95,43%	96,80%
10-15%	97,50%	97,50%	98,00%	98,50%
15-20%	96,32%	95,09%	95,71%	95,09%
20-25%	98,09%	95,54%	95,54%	98,73%
25-30%	96,49%	85,09%	91,23%	96,49%
30-35%	93,98%	77,11%	85,54%	98,80%
35-40%	86,57%	74,63%	82,09%	94,03%
40-50%	74,47%	63,83%	64,89%	81,91%
Más del 50%	73,33%	67,50%	47,50%	65,83%

Tabla 12. Precisión (micropromediada, *microaveraged*) de los cuatro identificadores.

Al micropromediar los resultados se comprueba que *blindLight* es sustancialmente superior al sistema de *XEROX* con más de un 25% de ruido; apreciablemente superior a *TEXTCAT* también a partir de ese mismo punto y sustancialmente a partir de un 40%. Al compararlo con *Acquaintance* las diferencias son inapreciables hasta un 30% de ruido, el rendimiento es peor entre un 30 y un 50% y sólo es materialmente superior con más de un 50% de ruido en el texto.

El análisis de estos resultados muestra que en términos absolutos *blindLight* es ligeramente superior al resto de técnicas cuanto mayor es la cantidad de ruido en el texto. No obstante, al macropromediar los resultados se comprueba que aunque, efectivamente, la técnica del autor es capaz de identificar correctamente un número mayor de documentos en presencia de ruido existen importantes diferencias de precisión entre los distintos idiomas[1] por lo que, tampoco en este experimento, se trata de la técnica más efectiva aunque los resultados son muy parejos hasta niveles de ruido del 30% y sólo *Acquaintance* se muestra claramente superior a *blindLight* a partir del 35%.

No obstante, como ya se dijo anteriormente parámetros importantes de los sistemas de referencia tales como la cantidad y la naturaleza del texto utilizada para construir los modelos de los otros sistemas de identificación son desconocidos por el autor. Sería interesante determinar si entrenando sobre otros tipos de documentos, empleando más texto[2] o utilizando otros tamaños de *n*-grama[3] mejoraría la precisión y en qué medida. Sin embargo, el autor considera que no es absolutamente necesario llegar en este trabajo a ese

[1] *blindLight* alcanzó una precisión de 100% y 94% en inglés y francés frente al 86,26% y 75,4% de castellano y alemán.

[2] Recuérdese que este identificador *blindLight* utilizó alrededor de 11 Kbytes por idioma

[3] El identificador *blindLight* aquí descrito empleó trigramas.

nivel de detalle y considera razonablemente argumentado que también en esta tarea la técnica que propone alcanza niveles de rendimiento comparables a los de métodos *ad hoc*.

Ruido en el texto	*blindLight*	Xerox	TEXTCAT	*Acquaintance*
0-5%	100,00%	99,17%	99,76%	99,52%
5-10%	100,00%	98,27%	97,96%	98,65%
10-15%	95,82%	98,94%	99,15%	99,36%
15-20%	95,22%	97,71%	98,25%	97,09%
20-25%	93,33%	98,08%	97,63%	98,89%
25-30%	94,49%	93,98%	95,60%	97,60%
30-35%	98,46%	90,26%	93,33%	99,23%
35-40%	79,50%	81,91%	88,21%	92,19%
40-50%	73,57%	77,86%	73,49%	83,55%
Más del 50%	54,67%	65,38%	44,38%	64,20%

Tabla 13. Precisión (macropromediada, *macroaveraged*) de los cuatro identificadores.

Los resultados macropromediados señalan que hasta un 30% de ruido los cuatro sistemas son análogos. Entre un 30 y un 35% de ruido la técnica del autor es apreciablemente mejor que la de *XEROX* y Cavnar y Trenkle (1994) y similar a *Acquaintance*. Con más de un 35% de ruido *blindLight* se comporta de manera peor que *XEROX* y *Acquaintance* aunque sólo la última es sustancialmente superior y sólo supera a *TEXTCAT* de manera sustancial con más de un 50% de ruido en el texto.

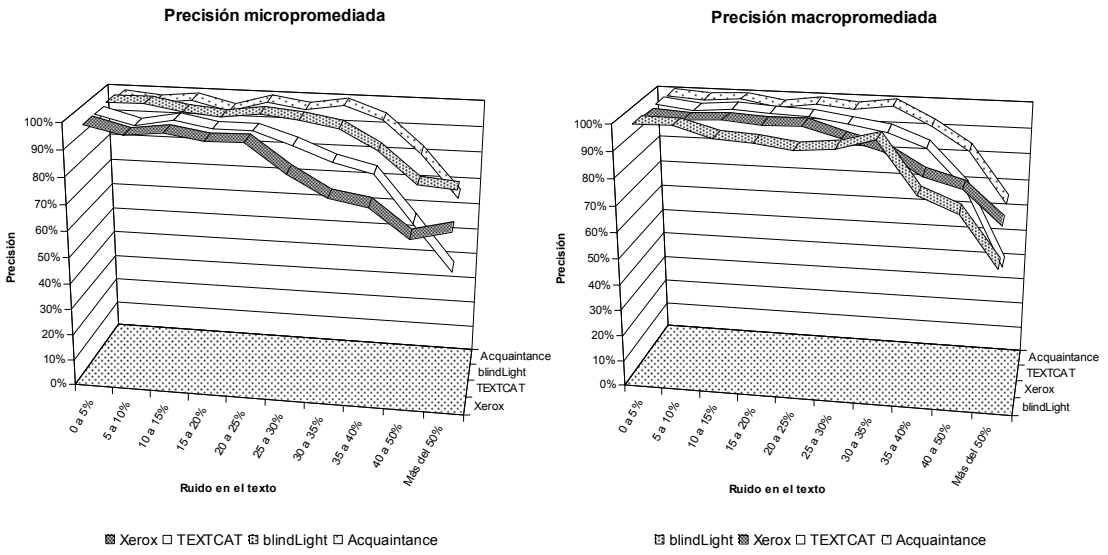

Fig. 85 Precisión de los identificadores en relación con el porcentaje de ruido.

5 Identificación de la autoría de un documento

Otro caso particular de categorización de documentos es la "atribución automática de autoría", es decir, la identificación del autor de un documento basándose en otros textos de dicho autor. Es necesario decir que no se considera que esta técnica sea especialmente fiable; Rudman (1998, p. 351), por ejemplo, afirma:

> *Los estudios no tradicionales de atribución de autoría –aquellos que emplean el ordenador, la estadística y la estilística– han tenido tiempo suficiente para superar cualquier "período transitorio" y entrar en una fase marcada por estudios sólidos, científicos y en constante progreso. Sin embargo, después de 30 años y 300 publicaciones no lo han hecho.*

En ese trabajo Rudman hace un repaso bastante exhaustivo tanto de las técnicas habituales como de las principales críticas a las mismas y a sus resultados. Esta situación es

análoga a la pugna existente entre los métodos tradicionales y estadísticos para clasificar lenguajes naturales y, como ésta, deberá ser resuelta por los investigadores involucrados en el campo. El autor tan sólo se ha aproximado a un problema clásico y bien estudiado dentro de la atribución de autoría: los denominados *Federalist Papers* (Artículos Federalistas).

Los *Federalist Papers* son una serie de 85 artículos publicados durante 1787 y 1788 en distintos periódicos del estado de Nueva York para convencer a los votantes de dicho estado sobre la necesidad de ratificar la futura constitución de los EE.UU. Dichos artículos aparecieron bajo el pseudónimo de Publius y fueron escritos por Alexander Hamilton, James Madison y John Jay. Posteriormente se llegó a un consenso sobre la autoría de cada artículo a excepción de 12, sobre los cuales sólo se estaba de acuerdo en que eran de Hamilton o de Madison.

Estos doce artículos son los conocidos como *disputed Federalist Papers* (los Artículos Federalistas disputados) y han generado bastante bibliografía: Mosteller y Wallace (1964, citado por Fung 2003) concluyeron, por métodos estadísticos, que los doce artículos en disputa eran obra de Madison. Desde entonces otros investigadores han empleado diversas técnicas[1] para alcanzar la misma conclusión y el autor también utilizó dicha colección como campo de prueba de la técnica *blindLight*.

Es preciso señalar que no se pretende hacer ninguna afirmación sobre la calidad de la técnica propuesta en el específico campo de la atribución de autoría puesto que Stamatos, Fakotakis y Kokkinakis (2001) advierten sobre las diferencias existentes entre los *Federalist Papers* y los textos que habitualmente se manejan en problemas de identificación de autoría. Simplemente se toma el problema de los artículos disputados como una tarea interesante de categorización.

El texto de los *Federalist Papers* fue obtenido en la Web[2] y procesado con *blindLight* empleando trigramas. Es necesario decir que el sitio "oficial" desde el que se descargaron muestra sólo once artículos de autoría dudosa, el duodécimo (atribuido a Madison en el primer sitio web) es el artículo número 58 según la *Emory School of Law*[3]. Así, la lista definitiva de artículos disputados sería la siguiente: 49 a 58, 62 y 63.

Artículo	Hamilton	Madison	Autor
49	0,286	0,370	Madison
50	0,282	0,352	Madison
51	0,287	0,370	Madison
52	0,283	0,369	Madison
53	0,298	0,368	Madison
54	0,257	0,375	Madison
55	0,309	0,350	Madison
56	0,255	0,384	Madison
57	0,307	0,367	Madison
58	0,279	0,361	Madison
62	0,291	0,360	Madison
63	0,279	0,368	Madison

Tabla 14. Valores *PiRoNorm* obtenidos por cada texto disputado al compararlo con ambos autores.

Para el conjunto de documentos escritos por cada autor en solitario se calculó el centroide y se empleó éste como vector representativo de la categoría. Como medida de

[1] Programación lineal (Bosch y Smith 1992, citado por Fung 2003), redes neuronales (Tweedie, Singh y Holmes 1994), algoritmos genéticos (Holmes y Forsyth 1994, citado por Buckland 1999), cadenas de Markov (Khmelev y Tweedie 2001) o *SVM* (Fung 2003).

[2] `http://thomas.loc.gov/home/histdox/fedpapers.html`

[3] `http://www.law.emory.edu/FEDERAL/federalist/`

similitud entre los documentos disputados y la categoría se empleó la versión normalizada de *PiRo* (véase pág. 118) y se obtuvieron los resultados que se muestran en Tabla 14 y que permiten atribuir a Madison todos los textos en disputa, lo cual está de acuerdo con las teorías más aceptadas.

6 Filtrado de correo no deseado (*spam*)

El problema del correo no solicitado fue anticipado con gran antelación por Postel (1975) y Peter Denning (1982) señaló la necesidad de investigar no sólo métodos de generar información sino también técnicas que permitan *"controlar y filtrar la información que llega a las personas que deben usarla"*. Cranor y Lamacchia (1998) determinaron que, en 1997, alrededor del 10% del correo recibido en una red corporativa era *spam*. Esta cifra no ha hecho sino aumentar; Whitworth y Whitworth (2004) analizan el trasfondo técnico, legal y social del correo no solicitado y presentan datos alarmantes: aproximadamente el 30% del correo entrante de cada usuario y más del 50% del correo transmitido es *spam*.

La solución del problema no es sencilla; después de todo, al desarrollarse técnicas que filtran mejor este correo basura sus autores envían más aún. Por lo que es muy probable que se sigan utilizando sistemas de categorización automática, bien sea en los servidores de difusión (*relaying*), en los de correo o en las aplicaciones cliente (véase Fig. 86).

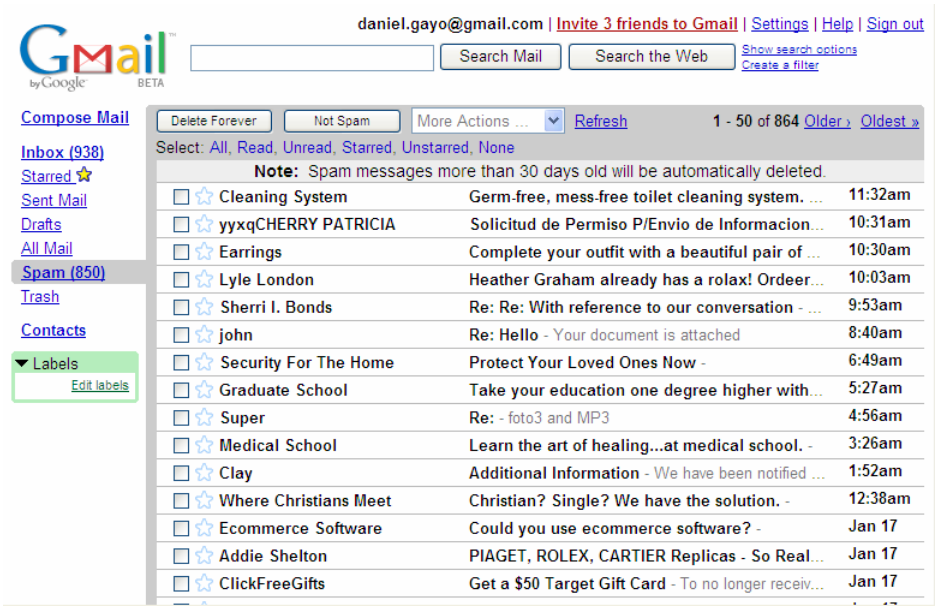

Fig. 86 Correo no solicitado filtrado automáticamente por *Gmail*.

Prácticamente todos los métodos de categorización revisados al comienzo del capítulo se han probado con el *spam*: Sahami *et al.* (1998), Pantel y Lin (1998) y Androutsopoulos *et al.* (2000a) utilizaron categorizadores bayesianos, Drucker, Wu y Vapnik (1999) *SVM's*, Androutsopoulos *et al.* (2000b) usaron *Memory Based Learning* y Zhang y Yao (2003) el modelo de entropía máxima. Según Zhang, Zhu y Yao (2004) *SVM's*, *boosting* y el modelo de máxima entropía son mucho mejores que los categorizadores bayesianos, una de las técnicas mejor consideradas para filtrar *spam* (Androutsopoulos *et al.* 2000a).

A fin de evaluar la utilidad de *blindLight* en esta misión, se emplearon dos *corpora* de correo utilizados en algunos de los trabajos anteriores: `ling-spam`[1] y `spamassassin`[2]. Existen otras colecciones de prueba pero para proteger la privacidad de los usuarios "donantes" no se ofrecen como texto plano sino codificadas haciéndolas inútiles para nuestros propósitos.

La colección `ling-spam` contiene 2.142 mensajes legítimos procedentes de una lista de correo y 481 mensajes no deseados, todos ellos en inglés y sin las correspondientes cabeceras. La colección `spamassassin` consta de 4.150 mensajes legítimos y 1.897 mensajes no deseados, también en inglés e incluyendo cabeceras. La primera colección está dividida en 10 partes para llevar a cabo validación cruzada mientras que la segunda fue dividida por el propio autor.

Para evaluar el rendimiento de la técnica propuesta en la categorización de correo no deseado se emplearán las mismas medidas que Androutsopoulos *et al.* (2000): la exactitud y tasa de error ponderadas, la razón de coste total (*total cost ratio* o *TCR*) así como la precisión y exhaustividad.

Si N_{legit} y N_{spam} es el número de mensajes legítimos y no deseados respectivamente y $n_{Y \to Z}$ es el número de mensajes de la categoría Y asignados a la categoría Z donde Y y Z pueden ser *legit* o *spam* entonces la exactitud (*accuracy*) y la tasa de error serían:

$$Acc = \frac{n_{legit \to legit} + n_{spam \to spam}}{N_{legit} + N_{spam}} \qquad Err = \frac{n_{legit \to spam} + n_{spam \to legit}}{N_{legit} + N_{spam}}$$

Esta definición de error y exactitud otorgan la misma importancia a categorizar un documento legítimo como *spam* que el caso contrario cuando el primer suceso es λ veces más costoso[3] por lo que es necesario ponderar estas medidas (Androutsopoulos *et al.* 2000):

$$WAcc = \frac{\lambda \cdot n_{legit \to legit} + n_{spam \to spam}}{\lambda \cdot N_{legit} + N_{spam}} \qquad WErr = \frac{\lambda \cdot n_{legit \to spam} + n_{spam \to legit}}{\lambda \cdot N_{legit} + N_{spam}}$$

Los valores comunmente asignados a λ son 1, 9 y 999 que se corresponderían a tres escenarios (Androutsopoulos *et al.* 2000): (1) marcar los mensajes no solicitados, (2) enviar una notificación al remitente y (3) borrar los mensajes. Quizás fuese más razonable reemplazar el segundo escenario por uno más plausible como mover los mensajes a una carpeta específica; después de todo, los remitentes de auténtico *spam* no emplean direcciones de correo reales.

No obstante, y a pesar de esta ponderación, la exactitud obtenida suele ser engañosamente alta y se hace necesaria una medida más "intuitiva" que permita comparar el filtro desarrollado con un sistema básico consistente en no filtrar ningún mensaje. La exactitud y tasa de error ponderadas para ese método serían las siguientes.

---

[1] `http://iit.demokritos.gr/skel/i-config/downloads/`

[2] `http://spamassassin.apache.org/publiccorpus/`

[3] Por ejemplo, resulta más sencillo eliminar un correo no deseado no capturado por el filtro que tener que explorar la carpeta de *spam* en busca de algún mensaje legítimo filtrado por error.

$$WAcc^b = \frac{\lambda \cdot N_{legit}}{\lambda \cdot N_{legit} + N_{spam}} \qquad WErr^b = \frac{N_{spam}}{\lambda \cdot N_{legit} + N_{spam}}$$

El cociente entre la tasa de error del sistema básico y del sistema a evaluar es la razón de coste total o *TCR* que tiene una interpretación muy simple: compara el esfuerzo dedicado a eliminar manualmente todo el *spam* recibido frente a eliminar el *spam* que aún pasa el filtro más recuperar correo legítimo categorizado como no solicitado. A mayor valor de *TCR* mejor rendimiento y, por otro lado, un valor de *TCR* inferior a la unidad significa que, en ese escenario en particular, es preferible dejar pasar todo el correo recibido que usar el filtro objeto de análisis.

$$TCR = \frac{WErr^b}{WErr} = \frac{N_{spam}}{\lambda \cdot n_{legit \to spam} + n_{spam \to legit}}$$

Los resultados obtenidos por *blindLight* con la colección `ling-spam` se muestran en Tabla 15. Por lo que respecta a exhaustividad y precisión la técnica propuesta por el autor fue capaz de capturar el 73,59% del *spam* de la colección con una precisión del 96,32% y tan sólo falla en el escenario más exigente (λ=999).

λ	WAcc	TCR
1	95,26%	3,51
9	99,02%	2,22
999	**99,58%**	**0,05**

Tabla 15. Rendimiento de *blindLight* categorizando la colección `ling-spam`.

Los resultados alcanzados al procesar la colección `spamassassin` fueron muy inferiores (véase Tabla 16). La exhaustividad ha sido algo mayor (77,95%) pero la precisión lograda ha sido mucho menor (84,81%). Por lo que respecta al *TCR* este es de 2,78 para λ=1 y de sólo 0,68 para λ=9, esto es, dada esta colección, este escenario y el método implementado sería preferible no filtrar el correo. Es preciso señalar, no obstante, que al procesar `spamassassin` los categorizadores bayesianos requieren vectores de entre 2000 y 3000 términos para conseguir mejorar al método básico (Zhang *et al.* 2004, p.10) no llegando a superar nunca un *TCR* de 2, mientras que con la colección `ling-spam` no precisaban más de 100 características (Androutsopoulos *et al.* 2000).

λ	WAcc	TCR
1	88,70%	2,78
9	**92,86%**	**0,68**
999	**93,91%**	**0,01**

Tabla 16. Rendimiento de *blindLight* categorizando la colección `spamassassin`.

La comparación con otras técnicas ha sido posible, hasta cierto punto, gracias a los trabajos de Androutsopoulos *et al.* (2000) y Zhang *et al.* (2004). Es necesario señalar que la técnica del autor no supera a ninguno de los otros métodos; sin embargo, en el caso de los categorizadores bayesianos y los que emplean *memory based learning* tan sólo hay diferencias en la exhaustividad, es decir, dichos métodos capturan más *spam* que *blindLight*. Estas diferencias son materiales en el caso de λ=1 y sólo apreciables para λ=9 y únicamente con el categorizador bayesiano. Métodos como *SVMs*, *boosting* o el de máxima entropía son muy superiores no sólo a *blindLight* sino también a los categorizadores bayesianos y *MBL* (Zhang *et al.* 2004) aunque sólo se ofrecen datos para λ=9 y 999, no para λ=1.

Se pueden extraer dos conclusiones de estos experimentos. En primer lugar antes de aplicar *blindLight* de forma efectiva a la particular tarea de filtrar *spam* sería necesario determinar la forma de introducir un modo de ponderar de manera distinta ambas categorías. Por otro lado, analizando estos datos como un experimento de categorización más, esto es, centrándonos en el escenario de $\lambda=1$, tenemos que la utilización de *blindLight* como categorizador proporciona resultados próximos a los de los del método bayesiano y *MBL*.

7 Comparación de *blindLight* con otras técnicas de categorización

Para poder comparar el rendimiento de la técnica propuesta por el autor con otros métodos de categorización existentes se llevó a cabo una serie de experimentos sobre las colecciones `Reuters-21578`[1] y `OHSUMED`[2].

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5549"
NEWID="6">
  <DATE>26-FEB-1987 15:14:36.41</DATE>
  <TOPICS>
    <D>veg-oil</D><D>linseed</D><D>lin-oil</D><D>soy-oil</D><D>sun-oil</D>
    <D>soybean</D><D>oilseed</D><D>corn</D><D>sunseed</D><D>grain</D>
    <D>sorghum</D><D>wheat</D>
  </TOPICS>
  <PLACES>
    <D>argentina</D>
  </PLACES>
  <PEOPLE></PEOPLE>
  <ORGS></ORGS>
  <EXCHANGES></EXCHANGES>
  <COMPANIES></COMPANIES>
  <UNKNOWN>
    G   f0754 reuter f BC-ARGENTINE-1986/87-GRA   02-26 0066
  </UNKNOWN>
  <TEXT>
    <TITLE>
      ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
    </TITLE>
    <DATELINE>
      BUENOS AIRES, Feb 26 -
    </DATELINE>
    <BODY>
      Argentine grain board figures show crop registrations of grains,
      oilseeds and their products to February 11, in thousands of tonnes,
      showing those for futurE shipments month, 1986/87 total and 1985/86
      total to February 12, 1986, in brackets:
      Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total
      2,692.4 (4,161.0).
      Maize Mar 48.0, total 48.0 (nil).
      Sorghum nil (nil)
      Oilseed export registrations were:
      Sunflowerseed total 15.0 (7.9)
      Soybean May 20.0, total 20.0 (nil)
    </BODY>
  </TEXT>
</REUTERS>
```

Fig. 87 Un documento de la colección *Reuters-21578*.

La primera consta de una serie de artículos (véase Fig. 87) publicados por la agencia de prensa *Reuters* durante el año 1987 a los que se asignaron manualmente una o más "etiquetas" de una lista[3] de 135 posibles. En la literatura se han empleado distintas particiones de la colección en conjuntos de entrenamiento y prueba. Por tanto, para obtener resultados comparables con los obtenidos por Joachims (1997) y Dumais *et al.* (1998) se ha

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[2] `ftp://medir.ohsu.edu/pub/ohsumed`

[3] En realidad, la colección `Reuters-21578` proporciona 5 conjuntos de categorías (`Exchanges`, `Orgs`, `People`, `Places` y `Topics`). Sin embargo, se suelen emplear únicamente las categorías correspondientes a `Topics` que hacen referencia a asuntos económicos, por ejemplo: `crude`, `nat-gas` o `iron-steel`.

utilizado la misma partición que estos investigadores, la denominada ModApte que utiliza 9.603 documentos para entrenamiento, 3.299 para test y descarta el resto de la colección.

Bacterial Infections and Mycoses	C01
Virus Diseases	C02
Parasitic Diseases	C03
Neoplasms	C04
Musculoskeletal Diseases	C05
Digestive System Diseases	C06
Stomatognathic Diseases	C07
Respiratory Tract Diseases	C08
Otorhinolaryngologic Diseases	C09
Nervous System Diseases	C10
Eye Diseases	C11
Urologic and Male Genital Diseases	C12
Female Genital Diseases and Pregnancy Complications	C13
Cardiovascular Diseases	C14
Hemic and Lymphatic Diseases	C15
Neonatal Diseases and Abnormalities	C16
Skin and Connective Tissue Diseases	C17
Nutritional and Metabolic Diseases	C18
Endocrine Diseases	C19
Immunologic Diseases	C20
Disorders of Environmental Origin	C21
Animal Diseases	C22
Pathological Conditions, Signs and Symptoms	C23

Fig. 88 Las 27 categorías de enfermedades presentes en *MeSH*.

Por lo que respecta a la colección OHSUMED consta de 348.566 referencias extraídas de *MEDLINE* y compuestas por el título y/o el resumen de artículos publicados en revistas médicas entre 1987 y 1991. Cada documento de la colección tiene asignado uno más términos de la clasificación *MeSH* (*Medical Subject Headings*) que proporciona un vocabulario controlado que abarca aspectos tan diversos como anatomía, trastornos mentales, procedimientos o medicamentos. Joachims (1997) utilizó los 20000 primeros documentos correspondientes a 1991 que incluían resumen, empleando la primera mitad durante el entrenamiento y la segunda para la fase de prueba. En cuanto a las categorías se limitó a las 23 entradas de primer nivel correspondientes a enfermedades (véase Fig. 88) suponiendo que un documento pertenece a una categoría dada si tiene asociado al menos un término índice de dicha categoría (véase Fig. 89).

```
Abdominal Pain/ET; Adolescence; Adult; Aged; Aged, 80 and over;
Appendicitis/CO/*RI/US; Child; Female; Human; Leukocytes/*; Middle Age;
Predictive Value of Tests; Support, Non-U.S. Gov't; Technetium Tc 99m
Aggregated Albumin/*DU.
```

```
Abdominal Pain;C23.888.646.100
Abdominal Pain;C23.888.821.030
Adult;M01.060.116
Aged;M01.060.116.100
Aged, 80 and over;M01.060.116.100.080
Appendicitis;C06.405.205.099
Appendicitis;C06.405.469.110.207
Child;M01.060.406
Leukocytes;A11.118.637
Leukocytes;A15.145.229.637
Predictive Value of Tests;E05.318.780.800.650
Predictive Value of Tests;G03.850.520.445.800.650
Predictive Value of Tests;H01.548.832.672.500
Predictive Value of Tests;N05.715.360.780.700.640
Technetium Tc 99m Aggregated Albumin;D02.691.825.375
Technetium Tc 99m Aggregated Albumin;D12.776.034.900
```

Fig. 89 Términos asociados a un documento de la colección *OHSUMED* y categorías *MeSH* correspondientes.

Joachims (1997), Dumais *et al.* (1998) y otros investigadores citados por Sebastiani (2002) emplearon como indicador del rendimiento de las técnicas analizadas el *breakeven*[1], aquel punto en el cual precisión y exhaustividad son iguales; la definición de precisión y exhaustividad en este contexto se presenta en las siguientes ecuaciones:

$$exhaustividad = \frac{categorías\ correctamente\ asignadas}{total\ de\ categorías\ correctas}$$

$$precisión = \frac{categorías\ correctamente\ asignadas}{total\ de\ categorías\ asignadas}$$

Así, al llevar a cabo los experimentos con *blindLight* se obtuvo, para cada documento, una lista ordenada de categorías comenzando por las más fuertemente vinculadas y terminando con las menos relacionadas con el documento. A partir de estos datos se extrajeron los correspondientes a precisión y exhaustividad que fueron "interpolados" (Chakrabarti 2003, p. 55) y, posteriormente, micropromediados. Una vez obtenidos estos últimos se aplicó interpolación lineal a aquellos pares de valores que delimitaban el valor de *breakeven* buscado.

Los resultados obtenidos por la técnica del autor en ambas colecciones se muestran en Tabla 17 y Tabla 18 junto con los alcanzados por Joachims (1997) y Dumais *et al.* (1998). La Tabla 19 permite comparar *blindLight* con toda una serie de técnicas aplicadas para categorizar la partición `ModApte` de la colección `Reuters-21578`. A la vista de tales resultados puede concluirse que, en general, *blindLight* es capaz de alcanzar resultados análogos a los de Rocchio, categorizadores bayesianos o árboles de decisión, resultados próximos aunque apreciablemente inferiores a los de *k*-vecinos y sustancialmente inferiores a los obtenidos con *SVMs*.

	blindLight	Bayes (i)	Bayes (ii)	Redes Bayes	Rocchio	*Findsim*	C4.5	Árboles de decisión	k-NN	SVM (poly) *d*=4	SVM (RBF) γ=0,8	SVM (lineal)
earn	94,5%	95,9%	95,9%	95,8%	96,1%	92,9%	96,1%	97,8%	97,3%	98,4%	**98,5%**	98,0%
acq	**99,3%**	91,5%	87,8%	88,3%	92,1%	64,7%	85,3%	89,7%	92,0%	95,2%	95,3%	93,6%
money-fx	51,7%	62,9%	56,6%	58,8%	67,6%	46,7%	69,4%	66,2%	**78,2%**	74,9%	75,4%	74,5%
grain	62,9%	72,5%	78,8%	81,4%	79,5%	67,5%	89,1%	85,0%	82,2%	91,3%	91,9%	**94,6%**
crude	76,3%	81,0%	79,5%	79,6%	81,5%	70,1%	75,5%	85,0%	85,7%	88,9%	**89,0%**	88,9%
trade	**95,5%**	50,0%	63,9%	69,0%	77,4%	65,1%	59,2%	72,5%	77,4%	77,3%	78,0%	75,9%
interest	39,6%	58,0%	64,9%	71,3%	72,5%	63,4%	49,1%	67,1%	74,0%	73,1%	75,0%	**77,7%**
ship	52,2%	78,7%	85,4%	84,4%	83,1%	49,2%	80,9%	74,2%	79,2%	86,5%	**86,5%**	85,6%
wheat	38,1%	60,6%	69,7%	82,7%	79,4%	68,9%	85,5%	**92,5%**	76,6%	85,9%	85,9%	91,8%
corn	31,3%	47,3%	65,3%	76,4%	62,2%	48,2%	87,7%	**91,8%**	77,9%	85,7%	85,7%	90,3%
μpromedio	77,7%	72,0%	75,2%	80,0%	79,9%	61,7%	79,4%	¿?	82,3%	86,2%	86,5%	87,0%

Tabla 17. Comparación de *blindLight* con otras técnicas al categorizar la colección *Reuters-21578*.

Se muestran los resultados para las 10 categorías más frecuentes y micropromediados para todas las categorías. Los datos de *Bayes* (i), *Rocchio*, *C4.5*, *k-NN*, *SVM* (*poly* y *RBF*) pertenecen a Joachims (1997), el resto a Dumais *et al.* (1998).

	blindLight	Bayes	Rocchio	C4.5	k-NN	SVM (poly) *d*=4	SVM (RBF) γ=1,0
Pathological Conditions, Signs and Symptoms	**83,9%**	52,7%	50,8%	47,6%	53,4%	58,2%	58,1%
Cardiovascular Diseases	69,6%	72,4%	70,1%	70,5%	72,6%	77,3%	**77,6%**
Immune System Diseases	55,0%	61,7%	58,0%	58,8%	66,8%	73,2%	**73,5%**
Neoplasms	**77,0%**	63,6%	64,1%	58,7%	67,2%	70,6%	70,7%
Digestive System Diseases	33,5%	65,3%	59,9%	59,0%	67,1%	73,7%	**73,8%**
Micropromedio	54,0%	57,0%	56,6%	50,0%	59,1%	65,9%	66,1%

Tabla 18. Comparación de *blindLight* con otras técnicas al categorizar la colección *OHSUMED*.

Se muestran los resultados para las 5 categorías más frecuentes y micropromediados para todas las categorías. Todos los datos proceden de Joachims (1997).

[1] Tanto Joachims (1997) como Dumais *et al.* (1998) señalan que los datos fueron micropromediados.

Técnica	Rendimiento
Boosting	87,8%
SVM	85,9%
k-NN	84,0%
Redes neuronales	83,8%
Reglas decision	82,2%
Redes Bayes	80,0%
C4.5	79,4%
blindLight	**77,7%**
Bayes	75,7%
Rocchio	72,0%

Tabla 19. Comparación del rendimiento de *blindLight* con otras técnicas al categorizar la partición *ModApte* de la colección *Reuters-21578*. Los datos se han obtenido de Sebastiani (2002, p. 38).

8 Influencia del tamaño de los *n*-gramas en la categorización

Todos los experimentos descritos hasta el momento fueron llevados a cabo empleando vectores de 3-gramas. En el capítulo anterior se llevó a cabo un experimento para evaluar la influencia del tamaño de los *n*-gramas sobre los resultados de la clasificación automática y en éste se ha hecho lo propio para determinar la influencia sobre el rendimiento del categorizador. Para ello se repitió el experimento descrito para la colección OHSUMED con 2-, 3- y 4-gramas.

Los resultados obtenidos se muestran en Tabla 20 y de ellos se desprende que el rendimiento al emplear vectores de bigramas en tareas de categorización no es adecuado al compararlo con los resultados obtenidos con vectores de 3- o 4-gramas. Sin embargo, para estos dos últimos tamaños no existen diferencias apreciables por lo que sería recomendable emplear trigramas en tareas de categorización puesto que tanto el tiempo de procesamiento como el espacio de almacenamiento necesarios son mucho menores.

Baste decir para terminar que se ha mostrado cómo la técnica propuesta por el autor puede efectivamente aplicarse al problema de la categorización de texto libre y obtener resultados análogos a los de muchos de los métodos convencionales.

	2-gramas	3-gramas	4-gramas
Pathological Conditions, Signs and Symptoms	44,03%	83,89%	**86,59%**
Cardiovascular Diseases	32,01%	**69,64%**	66,89%
Immune System Diseases	27,22%	**54,98%**	53,36%
Neoplasms	**86,95%**	76,95%	76,40%
Digestive System Diseases	**39,73%**	33,54%	37,39%
Micropromedio	51,45%	53,96%	**54,42%**
Macropromedio (top 5)	45,99%	63,80%	**64,13%**

Tabla 20. Comparación del rendimiento de *blindLight* empleando distintos tamaños de *n*-grama.