Un *Framework* para la Gestión Documental en las Administraciones Públicas

E.F. Combarro, E. Montañés, I. Díaz, R. Cortina, P. Alonso, J. Ranilla

Abstract: En este trabajo se presentan técnicas de agrupamiento de información que ayudan a gestionar la información de forma eficaz dado que aumentan la velocidad de recuperación, o facilitan su gestión. Dado que la aplicación de estas técnicas es en sí misma costosa, se han introducido sistemas de computación paralela y bibliotecas de alto rendimiento para mejorar la eficiencia de estos métodos.

1 Introducción

La explotación eficiente y efectiva de grandes colecciones de información proporciona importantes beneficios, tanto científicos como económicos ([23]). Si bien los medios tecnológicos actuales, especialmente informáticos, posibilitan la obtención y el almacenamiento de grandes cantidades de información, la llamada *Sociedad de la Información* está siendo superada por la necesidad de nuevos métodos capaces de procesar la información de forma eficiente y eficaz. Esto es particularmente cierto en el caso de las Administraciones Públicas, que tienen la obligación de gestionar de forma efectiva, eficiente y segura grandes cantidades de información, mucha de ella en forma de documentos de texto.

En general, la información es un conjunto de unidades o elementos base (por ejemplo, documentos de texto, imágenes o videos) caracterizados por un gran número de aspectos (también llamados atributos, características, propiedades o, en inglés, *features*) de los cuáles probablemente sólo una reducida parte sea representativa o necesaria para la correcta explotación de la información ([19], [23]). Y puesto que el éxito en el tratamiento de la información está inexorablemente condicionado por el punto de partida, se hace necesario estudiar, desarrollar y evaluar métodos de reducción de atributos que resulten efectivos cuando se tratan entidades caracterizadas por un número elevado de propiedades.

Pero no sólo el elevado número de características condiciona el éxito de un sistema de clasificación/recuperación, también influyen el tamaño del problema (número de documentos), la ambigüedad de las preguntas (*queries*), etc. En este sentido, los algoritmos de agrupamiento son una herramienta de contrastada eficacia. Eficacia en términos de la calidad del resultado obtenido (el agrupamiento puede, debe, detectar similitudes entre objetos) y de la velocidad en la recuperación (solamente es necesario realizar la búsqueda sobre los representantes de cada grupo).

Las técnicas de reducción de la dimensión tienen como objetivo reducir el espacio de búsqueda (número de atributos o características) y, consecuentemente, el tiempo necesario para efectuar los procesos de la recuperación subsiguientes, como es el caso del agrupamiento, así como el espacio de almacenaje. Sin embargo, la propia aplicación de las técnicas de reducción es temporalmente costosa, y el resto de procesos, aun habiendo reducido el espacio de búsqueda, también lo son. Para disminuir el coste temporal asociado a estos procesos, el paralelismo es probablemente la mejor opción.

Por tanto, la optimización de la eficiencia computacional en la recuperación y/o clasificación de ingentes cantidades de documentos será abordada con técnicas de programación paralela y usando bibliotecas de alto rendimiento. En lo que respecta al paralelismo, se ha optado por emplear el modelo de programación de memoria distribuida sobre arquitecturas *NOW* (*Network Of Workstations*) ([3]), siendo este tipo de arquitecturas un recurso habitual en las Administraciones Públicas.

La estructura del trabajo es la siguiente. En la sección 2 se muestra la arquitectura genérica del sistema, en la sección 3 se detallan algunas de las técnicas de reducción utilizadas, en la sección 4 se indican las técnicas de *clustering* utilizadas para recuperar información. La sección 5 enumera los métodos de clasificación que han sido propuestos y evaluados para el proceso de la clasificación de documentos. La sección 6 define las arquitecturas paralelas utilizadas. Finalmente, en la sección 7 se proporciona un ejemplo de una aplicación real, y en la sección 8 las principales conclusiones de este trabajo.

2 Arquitectura del Sistema

El *framework* que presentamos en este documento se ha diseñado teniendo en cuenta las similitudes y diferencias de las dos principales tareas que un sistema de gestión documental debe abordar. Por un lado, deberemos ser capaces de **recuperar información** relevante de entre la almacenada en el sistema. La recuperación propiamente dicha se lleva a cabo por medio de consultas que los usuarios introducen en el sistema. Cuando una consulta es introducida, el sistema determina cuáles son los documentos relevantes y devuelve la información correspondiente a esos documentos. Por otro, querremos tener **clasificada** aquella información que, debido a que presenta un mayor grado de estructuración (por ejemplo, proporciona etiquetas que informan de su contenido) es susceptible de ser organizada en algún tipo de jerarquía.

Ambas tareas presentan diferencias en cuanto a los métodos que se aplican y las técnicas de reducción que es posible utilizar, pero tienen muchas etapas comunes, especialmente en lo relativo al procesado previo de los documentos y a la forma que en que la información es representada en el sistema para su tratamiento posterior. Tanto los documentos como las consultas se caracterizan mediante vectores (que se conocen como *bag of words* ([19]) cuyos elementos son valores de peso asociados a cada uno de los términos que se encuentran en la colección. De esta manera, la dimensión de los vectores es el número de raíces (*stems*) de palabras distintas que aparecen en la colección y cada una de las dimensiones se corresponde con un término o *stem*. La representación conjunta de todos los documentos de una colección se realiza por medio de una matriz llamada matriz *término/documento* ([21]). Existen diversas formas de cuantificar la relevancia de las palabras de un documento. La más simple consiste en asignar a cada término un peso de **1**

en el caso de que se encuentre en el documento, y un peso de **0** en caso contrario. Otra posibilidad es considerar la frecuencia del término, es decir, el número de veces que aparece en el documento. Se suele denotar por *tf* (*term frequency*). Estas dos funciones de peso tienen la ventaja de que son fáciles de interpretar y de calcular.

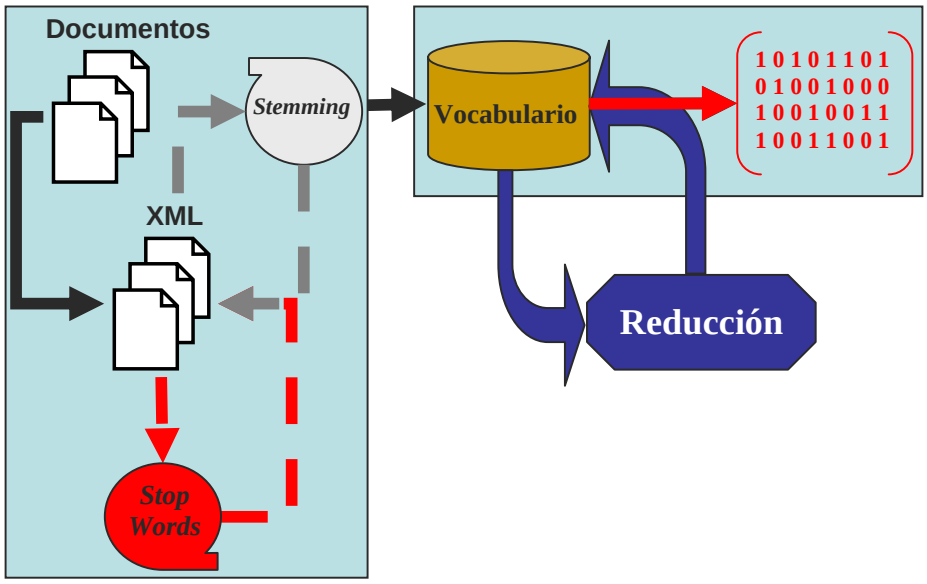

Figura 1: Etapas en la creación de la matriz término/documento

Las etapas necesarias para obtener la matriz término/documento se presentan de forma gráfica en la **Figura 1** y son las siguientes ([23]):

- En primer lugar, los documentos se **convierten** del formato original (*txt*, *doc*, *pdf*,…) a una representación común, como por ejemplo el lenguaje ***XML***.
- Se eliminan las **palabras vacías** como artículos, pronombres, adverbios, etc., que no resultan útiles para la clasificación de los documentos.
- De las palabras restantes se obtienen las raíces o *stems* que definen el vocabulario de la colección. Para abordar este proceso, conocido como ***stemming***, existen distintos algoritmos para diversos idiomas.
- A continuación se efectúa la **reducción del vocabulario**, que dependerá del tipo de aplicación (clasificación o recuperación) que se vaya a realizar.

Para este último proceso existen numerosas técnicas que pasamos a detallar en las siguientes secciones.

3 Técnicas de Reducción de Dimensionalidad

Habitualmente, en las colecciones de documentos hay palabras que se repiten en la mayor parte de ellos. Son términos que no son significativos a la hora de distinguir un documento de otros. Por otro lado, los términos que aparecen muchas veces en pocos documentos sí pueden servir para este propósito.

La reducción mediante **selección de términos** consiste en obviar aquellas palabras que aparecen en la mayor parte de los documentos de la colección, con el fin de mantener aquellas que pueden ser más útiles para las búsquedas ([4]), es decir, más relevantes, o si se prefiere, de mayor peso. Para ello se realiza una valoración de la relevancia de cada una de las palabras del vocabulario de la colección, se ordena el vocabulario de acuerdo a ese valor, y, finalmente, se descarta un porcentaje arbitrario de las palabras con menor relevancia. Las funciones más utilizadas son ***TFIDF*** **y** ***TFV*** ([4], [22]).

Por otro lado, las técnicas de **extracción de términos** permiten combinar la información presente en varios atributos formando nuevos atributos sintéticos que conservan la semántica de los términos agrupados. Por ejemplo, ***Latent Semantic Index***, o ***LSI*** ([6], [16]), es un método que permite determinar la estructura semántica latente de la relación entre documentos y términos con el fin de superar las deficiencias de los sistemas de recuperación de información basados en la similitud por coincidencia de términos. Existen varias técnicas para determinar dicha estructura semántica, pero la más común está basada en la descomposición *SVD* (*Singular Value Decomposition*) de la matriz de términos y documentos de la colección.

LSI es un método que tiene un coste computacional muy elevado, asociado a la descomposición *SVD*. Por otro lado, puede resultar difícil estimar el valor de $\boldsymbol{k}$ más adecuado. La reducción de la dimensión mediante **proyección aleatoria** ([1], [4]) está considerada como una alternativa a *LSI* con menor coste computacional y una pérdida de información razonable. La proyección aleatoria se basa en el lema de *Johnson-Lindenstrauss* ([1]). Este resultado asegura que cualquier conjunto de vectores de un espacio euclídeo puede ser proyectado sobre otro espacio de menor dimensión, de tal forma que las distancias entre pares de vectores en el espacio original se mantienen de manera aproximada en el nuevo espacio.

En el caso de la clasificación de documentos, algunos de éstos se presentan previamente **categorizados** o **etiquetados**, por lo que se dispone de una información adicional a la hora de señalar los términos relevantes: la categoría de los documentos en los que aparecen. Esto permite utilizar, además de los métodos mencionados anteriormente, técnicas especificas de selección de términos que aprovechan esta información. Las más utilizadas están basadas en **medidas de relevancia** que asignan un valor numérico a cada uno de los términos y permiten, por tanto, realizar una reducción del vocabulario al quedarse sólo con aquellos términos más relevantes ([15], [25]). Algunas de las medidas más utilizadas son la ganancia de información o la CET (*cross entropy for text*). En trabajos previos hemos identificado algunas nuevas familias de medidas (por ejemplo las llamadas *medidas lineales*) que son especialmente efectivas y eficaces en la tarea de la reducción del vocabulario ([5]).

4 Recuperación de Información por medio de *Clustering*

En los sistemas de recuperación de información basados en *bag of words*, la recuperación se lleva a cabo por comparación entre las consultas y los documentos de la colección, de tal forma que, dada una consulta, se retornan aquellos documentos similares a ella. Existen distintas formas de determinar la similitud entre las consultas de usuario y los documentos a partir de sus vectores representativos ([19], [20], [22]). Considérense $\boldsymbol{d}$ y $\boldsymbol{q}$

vectores de dimensión ***n*** que representan un documento y una consulta respectivamente. La técnica más utilizada consiste en obtener el coseno del ángulo que forman mediante

$$\cos(\theta) = \frac{\langle d, q \rangle}{\|d\| \cdot \|q\|} \qquad (1)$$

La similitud entre la consulta y el documento será entonces mayor cuanto mayor sea el valor de *(1)*, o de manera equivalente, cuanto menor sea el ángulo que forman.

Normalmente, ya que el número de documentos existentes es muy elevado, resulta muy costoso realizar la comparación entre la consulta del usuario y todos los documentos de la colección. Por eso, se puede llevar a cabo un agrupamiento previo de los documentos en conjuntos de semejante contenido de forma que se reduce el número de comparaciones necesarias al comparar las consultas sólo con los representantes de cada uno de los grupos. Este proceso se presenta de forma gráfica en la **Figura 2**.

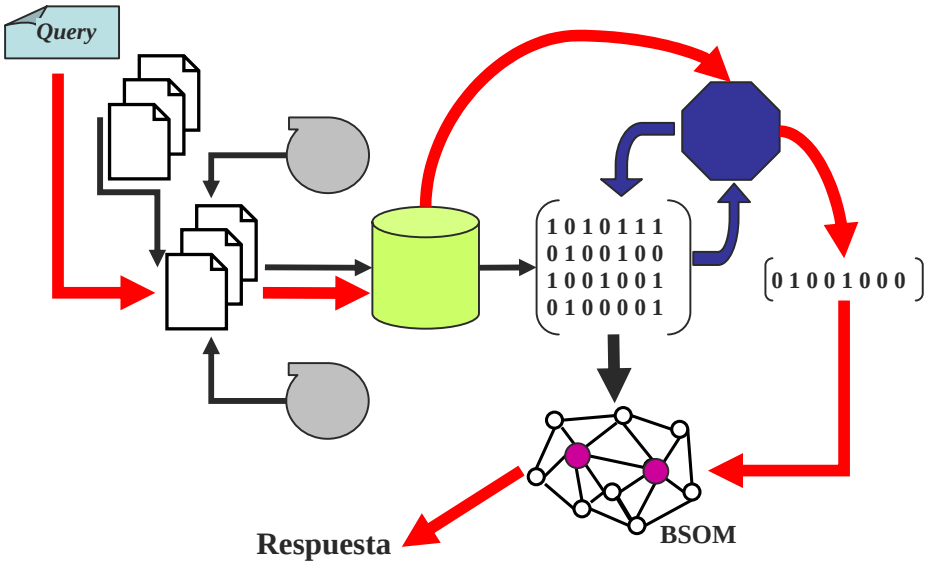

Figura 2: Recuperación de información con agrupamiento previo de documentos

Para llevar a cabo este agrupamiento o ***clustering*** existen numerosas técnicas en la literatura. En nuestro *framework* se encuentran disponibles algunas de las más populares, como pueden ser ***k-means*** o ***SOM***.

El algoritmo *k-means* ([13]) es un método diseñado para particionar en ***k*** grupos los objetos de entrada, basándose en las características de éstos. El objetivo es conseguir ***k*** grupos o *clusters*, minimizando la varianza interna de los grupos.

Los mapas autoorganizados o *SOM* (*Self-Organizing Map*) ([9]) son un modelo de red neuronal que es capaz de proyectar datos de entrada de grandes dimensiones sobre un *espacio* de dos dimensiones. Esta proyección da como resultado un mapa que puede ser de utilidad para la detección y el análisis de características del espacio de los datos de entrada. Los mapas autoorganizados han sido aplicados con éxito en diversas disciplinas, incluyendo la Clasificación Documental ([10], [12]) y la Recuperación de Información

([11]). Dos son los tipos fundamentales de *SOMs* en función del tipo de entrenamiento que reciben: a) Clásica, también llamada ***En Línea*** (*On-Line*), y b) ***Por Lotes*** (*Batch*). Si bien la convergencia, o velocidad de convergencia, a la solución de las redes *Batch* puede ser menos efectiva que en las *On-Line*, son las más eficientes cuando se usa el paralelismo, como es nuestro caso. Ambas versiones están disponibles en nuestro *framework*.

5 Métodos de Clasificación

Varios son los métodos de clasificación que han sido propuestos y evaluados para el proceso de la clasificación de documentos. Se describen a continuación los más utilizados:

- **Naive Bayes**. Este método consiste en utilizar la probabilidad conjunta de palabras y categorías para estimar la probabilidad de cada categoría condicionada a un documento. Para ello, se utiliza la fórmula de *Bayes* ([14]).
- **Árboles de Decisión**. Dentro del aprendizaje automático existen métodos que construyen árboles de decisión a partir de un conjunto de ejemplos. En un árbol de decisión, los nodos están etiquetados por un *test* relativo a los valores de los ejemplos en un atributo (o varios), y las hojas, por una de las categorías. La aplicación del árbol de decisión permite clasificar ejemplos no vistos anteriormente. Algunos de los métodos más utilizados para la construcción de árboles de decisión incluyen CART ([2]), C4.5 ([17]) y ARNI ([18]).
- **Máquinas de Vectores de Soporte**. Las máquinas de vectores de soporte (*Support Vector Machines* o SVM) se basan en el principio de minimización estructural de riesgo de la teoría de aprendizaje computacional. Este principio trata de encontrar una hipótesis h que garantice el menor error sobre un ejemplo no visto. Básicamente, este método construye un hiperplano que maximiza la separación entre categorías ([24], [8]). Este es el método que mejores resultados ofrece y el que por defecto se utiliza en nuestro sistema ([5], [7]).

6 Arquitecturas *NOW*

Las arquitecturas *NOW* presentan una serie de características que las hacen muy interesantes para sistemas de recuperación de información. Por un lado, son arquitecturas altamente escalables, por lo que se pueden adaptar fácilmente los recursos a las necesidades de la recuperación. Por otra parte, si el hardware subyacente es adecuado, se pueden alcanzar cotas de rendimiento muy elevadas. Además, pueden ser configuradas para que sus nodos sean utilizables para diversas tareas. A todo esto hay que añadir los bajos costes de adquisición y mantenimiento que suponen frente a otro tipo de arquitecturas.

En la **Figura 3** se muestran algunas de las posibles configuraciones de arquitecturas *NOW*. Los elementos sombreados se corresponden con aquella configuración específica hacia la que se ha orientado este *framework*. En este sentido, lo más destacable sea quizás el uso de *MPI*[1] (*Message Passing Interface*). *MPI* es una especificación estándar para bibliotecas de paso de mensajes entre diferentes procesos dirigida al diseño de algoritmos

[1] http://www-unix.mcs.anl.gov/mpi/

paralelos. Cabe mencionar también la elección del conocido sistema operativo *Linux* y de la paquete *software LTSP* (*Linux Terminal Server Project*) para la gestión de los nodos de la red. La eficacia y la estabilidad de todo este software están contrastadas. Pero además se trata de software de libre distribución, por lo que la mayor parte de los costes derivados del mismo se centran en el mantenimiento.

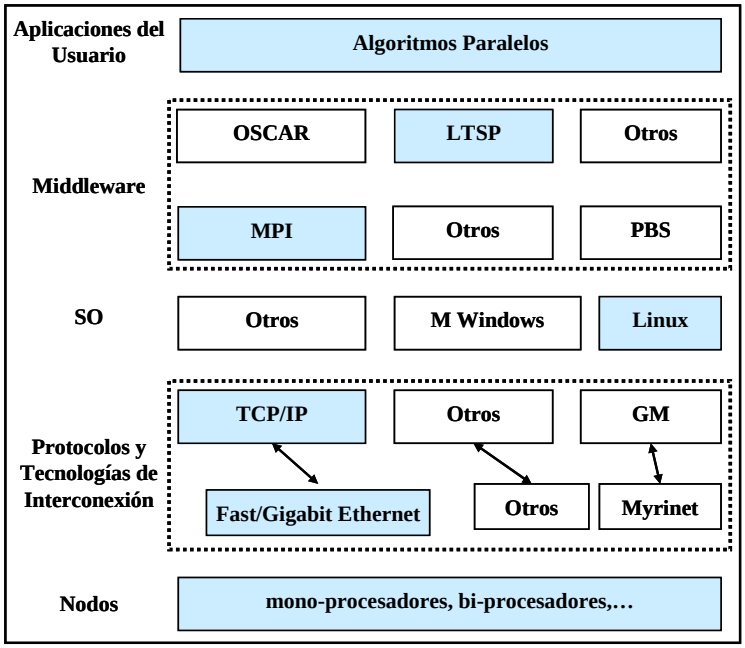

Figura 3: Arquitecturas *NOW*

La limitación, si es que se puede definir como tal al ser común a todos los clústeres de ordenadores, es la necesidad de nodos *front-end* potentes, de sistemas de almacenamiento rápidos y robustos, y de un buen sistema de interconexión entre los distintos elementos de la red.

7 Un Prototipo

Durante la realización de los proyectos de investigación *Desarrollo de Software Inteligente Basado en Aprendizaje automático aplicado a problemas reales de Ordenación y Clasificación* (SIBAOC) del Ministerio de Ciencia y Tecnología (TIC2001-3579, 2001-2004) y *Organización y Recuperación de Información Paralela e Inteligente* (ORIPI) del Ministerio de Educación y Ciencia (TIN2004-05920, 2004-2007) y, en colaboración con el **Excelentísimo Ayuntamiento de Gijón**, se desarrolló e implementó un prototipo de sistema de recuperación y clasificación de información basado en el *framework* propuesto en esta comunicación.

La base documental utilizada contenía actas de los plenos del Ayuntamiento de Gijón correspondientes a distintos años que, pese a contener información de interés que era necesario consultar frecuentemente, resultaban poco accesibles y difíciles de manejar.

Para la implementación del prototipo se llevó a cabo la conversión automática de la base documental de su formato original a la representación utilizada por nuestro sistema. Se adaptaron todos los métodos (eliminación de palabras vacías, *stemming*,…) a las particularidades del lenguaje español y al vocabulario específico utilizado en los documentos. Se realizaron estudios para determinar las técnicas más adecuadas para la reducción del vocabulario. Finalmente, se utilizaron redes neuronales tipo *SOM* entrenadas en modo *Batch* en una red de ordenadores con arquitectura *NOW* para realizar el agrupamiento de los documentos. El *interface* para la consulta de la base de datos documental se implementó como un servicio *Web* de sencillo uso y amigable al usuario.

8 Conclusiones

La gestión de grandes colecciones de documentos es una tarea de especial relevancia en la sociedad de la información y es un reto particularmente importante para las Administraciones Públicas. Dos de las tareas más importantes en esta gestión documental son, por un lado, la recuperación eficiente y efectiva de la información existente y, por otro, la clasificación automática de nuevos documentos presentados al sistema.

En la literatura existen gran cantidad de soluciones para cada una de las subtareas que constituyen cada una de las etapas de construcción de un sistema de gestión documental. En el presente documento hemos presentado un *framework* que permite integrar las mejores y más interesantes de estas soluciones para construir un sistema robusto, eficiente y eficaz que permita la clasificación y recuperación de la información disponible. Además, la posibilidad de usar arquitecturas de redes de ordenadores de bajo coste (*NOW*) es especialmente atractiva, puesto que proporciona las ventajas de la computación paralela sin incurrir en costes excesivos.

Futuras evoluciones de este *framework* podrían incluir nuevas tareas para la gestión documental, como por ejemplo la elaboración de resúmenes o la creación automática de taxonomías específicas de un campo de conocimiento. La integración de estas tareas en el sistema puede realizarse de forma relativamente sencilla, dado que las partes comunes en la gestión documental han sido realizadas de forma totalmente independiente de los procesos posteriores y cada subtarea constituye un módulo completamente independiente que puede ser reemplazado por otro o incluso reutilizado en otros sistemas.

9 Agradecimientos

La investigación y el *framework* presentado en este documento se han realizado en el marco de los proyectos *SIBAOC* (TIC2001-3579) y *ORIPI* (TIN2004-05920). También queremos agradecer la colaboración del Excelentísimo Ayuntamiento de Gijón que proporcionó una de las bases documentales que se utilizaron en el estudio.

10 Bibliografía

[1] D. Achlioptas: Database-friendly Random Projections. En *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (2001)

[2] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen: Classification and Regression Trees. *Chapman & Hall/CRC; 1 edition* (1984)

[3] E.F. Combarro, E. Montañés, I. Díaz, R. Cortina, P. Alonso, J. Ranilla: NoW Architectures, Dimensionality Reduction and Self-Organizing Maps for Information Retrieval. En *Lecture Series on Computer and Computational Science*, 7: 110-113 (2006)

[4] E.F. Combarro, E. Montañés, I. Díaz, R. Cortina, P. Alonso, J. Ranilla: Dimension Reduction Techniques for Information Retrieval using Self-Organizing Maps. En *Proceedings of the 11th Information Processing and Management of Uncertainty in Knowledge-Based Systems International Conference (IPMU 2006)* (2006)

[5] E.F. Combarro, E. Montañés, I. Díaz, J. Ranilla, R. Mones: Introducing a Family of Linear Measures for Feature Selection in Text Categorization. En *IEEE Transactions on Knowledge and Data Engineering*, 17(9): 1223-1232 (2005)

[6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman: Indexing by Latent Semantic Indexing. En *Journal of the American Society for Information Science*, 41(6): 391-407 (1990)

[7] I. Díaz, J. Ranilla, E. Montañés, J. Fernández, E.F. Combarro: Improving Performance of Text Categorisation by Combining Filtering and Support Vector. En *Journal of the American Society for Information Science and Technology (JASIST)*, 55(7): 579-592 (2004)

[8] T. Joachims: Text categorization with support vector machines: learning with many relevant features. En *Proceedings of the10th European Conference on Machine Learning (ECML-98)* (1998)

[9] T. Kohonen: Self-Organizing Maps. *Springer-Verlag* (2001)

[10] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela: Self-Organization of a Massive Document Collection. En *IEEE Transactions on Neural Networks*, 12(3): 574-585 (2000)

[11] K. Lagus: Text Retrieval Using Self-Organized Document Maps. En *Neural Processing Letters*, 15(1): 21-29 (2002)

[12] K. Lagus, J. Honkela, S. Kaski, T. Kohonen. Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration. En *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996)

[13] J.B. MacQueen: Some Methods for classification and Analysis of Multivariate Observations. En *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability* (1967)

[14] T. Mitchell: Machine Learning. *McGraw Hill* (1997)

[15] E. Montañés, I. Díaz, J. Ranilla, E.F. Combarro, J. Fernández: Scoring and Selecting Terms for Text Categorization. En *IEEE Intelligent Systems*, 20(3): 40-47 (2005)

[16] C.H. Papadimitriou, P. Raghavan, H. Tamaki: Latent Semantic Indexing: A Probabilistic Analysis. En *Proceedings of the ACM Conference on Principles of Database Systems* (1997)

[17] Quinlan, J.R.: Constructing decision trees in C4.5: Programs of Machine Learning. *Morgan Kaufman Publishers* (1993)

[18] J. Ranilla, A. Bahamonde: Fan: Finding accurate inductions. En *International Journal of Human Computer Studies*, 56(4): 445-474 (2002)

[19] G. Salton and M.J. McGill: An Introduction to Modern Information Retrieval. *McGraw-Hill* (1983)

[20] G. Salton, A. Wong, C.S. Yang: A Vector Space Model for Automatic Indexing. En *Communications of the ACM*, 18(11): 613-620 (1975)

[21] F. Sebastiani: Machine Learning in Automated Text Categorisation. *ACM Computing Survey*, 34(1): 1-47 (2002)

[22] B. Tang, M. Shepherd, E. Milios, M.I. Heywood: Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering. En *SIAM International Workshop on Feature Selection for Data Mining - Interfacing Machine Learning and Statistics* (2005)

[23] C.J. Van Rijsbergen: Information Retrieval (www.dcs.gla.ac.uk/Keith/Preface.html)

[24] V.N. Vapnik: The Nature of Statistical Learning Theory. *Springer-Verlag* (1995)

[25] T. Yang, J.P. Pedersen: A Comparative Study on Feature Selection in Text Categorisation. En *Proceedings of the14th International Conference on Machine Learning (ICML'97)* (1997)