Web Mining con Java

M. Daniela López De Luise *

Introducción

Este artículo es un extracto de la disertación realizada en el marco de las *Jornadas Tecnológicas* del mes de Agosto de 2005 y de las actividades conjuntas de la UP con el IEEE. Se presentarán los conceptos fundamentales de la actividad de Web Mining y luego algunas aplicaciones.

A. El Web Mining

Web Mining (WM) es una actividad que surge como extensión de una actividad denominada Data Mining (DM), que normalmente se aplica a grandes Bases de Datos (BD) para descubrir y extraer información desconocida previamente, no evidente y relevante. En el caso del WM, esta extracción no se hará desde una BD sino desde un conjunto de documentos almancenados en la Web y de un conjunto de Servicios Web o Web Services (WS).

Las tareas que conciernen a este tipo de actividad suelen ser:

- Detección de los recursos que potencialmente son útiles para la obtención de la información deseada (los recursos pueden ser Newsgroups, NewsLetters, Bases de Datos offline/online, documentos offLine/online, páginas html, etc).
- Selección automática de los mejores recursos según algún criterio.
- Preproceso o preparación de estos recursos para ser procesados de manera homogénea y consistente por un sistema automático.
- Detección de patrones de comportamiento o patrones de información de manera automática.
- Validación de estos patrones para ver si realmente manifiestan información no evidente y relevante.
- Interpretación del significado de estos patrones.

A continuación detallaremos por qué la Web es una BD especial y por qué las técnicas de DM pueden ser una solución plausible para poder procesar su información.

A.i.La Web como Base de Datos

La Web como Base de Datos es muy especial ya que tiene características muy peculiares:

1. Su tamaño es el mayor jamás abarcado por cualquier BD.

* Docente de la Facultad de Ingeniería - UP.

2. Es muy heterogénea en su contenido.
3. Es altamente dinámica.

Como consecuencia de ésto, surgen problemas específicos para los usuarios y proveedores de la información contenida en ella:

a) Los usuarios: tendrán mayor inconveniente en hallar la información que ellos consideran relevante para su necesidad actual.
b) Los proveedores: deberán aplicar técnicas más depuradas para aprender acerca de las necesidades y preferencias de los consumidores (que ahora serán realmente muchos), deberán generar alternativas plausibles de personalización de la información (a través de técnicas textuales y/o visuales muy precisas y a veces complejas) y deberán crear nueva información para explicar, expander y reusar convenientemente la información almacenada.

Ahora veremos cómo podríamos encarar la solución de estos problemas.

A.ii.El Data Mining en la Web

Actualmente existen muchas alternativas que intentan administrar la información contenida en la Web para compensar y solucionar los problemas a usuarios y proveedores. Algunas de estas alternativas son: técnicas tradicionales de Bases de Datos, procesamiento de Lenguaje Natural (denominada NLP por Natural Language Processing), técnicas específicas de Information Retrieval (IR) , y el mismo Web Mining entre otras.

En el caso del Web Mining (WM), para proveer solución a estos problemas se especializa en ciertas *categorías de Mining* según el tipo de información a procesar y extraer:

a.Web content mining

Procesa propiamente sobre el contenido de la Web, ya sean sistemas propietarios colgados a través de una interface apropiada, WS o librerías digitales disponibles en la Web. En términos generales se pueden ver todos estos como conjuntos de datos con distintos niveles de estructuración.

b.Web structure mining

Trabaja la información contenida en la topología de links e hiperlinks. Normalmente tiene sensibilidad suficiente para clasificar páginas y sites de manera distinta.

c.Web usage mining

Estudia la información que generan las sesiones de los internautas: cookies, queries, clicks, información enviada en formularios Web, logs de error y de acceso, etc. Este tipo de estudios ha provocado cierto resquemor en la comunidad ya que puede considerarse a veces como una violación a la privacidad y un atentado potencial contra la seguridad de la información. Esto ha ocasionado el surgimiento de ciertas normativas y propuestas.

Como puede apreciarse, las distintas variantes estudian la información disponible totalmente y por lo tanto pueden resolver los problemas no sólo del usuario sino también de los proveedores.

Existen otras razones para usar WM además del amplio alcance de sus técnicas. Por un lado, el gran crecimiento físico (Fig. 1) y lógico (Fig. 2) de la Web desde sus comienzos hace pensar que existe una tendencia abrumadora a colocar contenidos en internet.

Fig. 1. Crecimiento de la cantidad de Hosts en el comienzo de internet

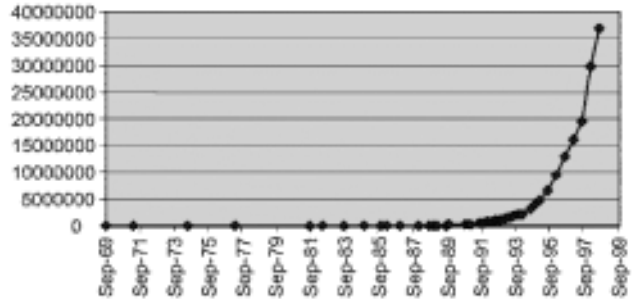
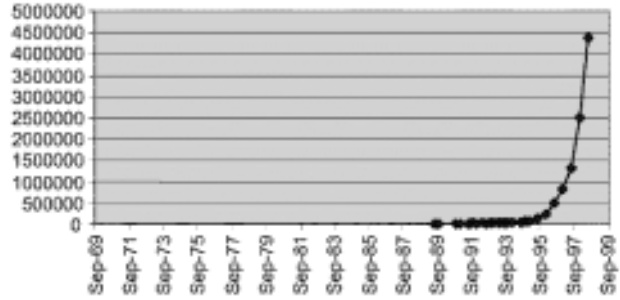


Fig. 2. Crecimiento en la cantidad de dominios en el comienzo de internet



Como se ve en las figuras, el crecimiento ha sucedido en muy poco tiempo y su crecimiento exponencial es altamente preocupante, no sólo por la posibilidad de colapsar la estructura global y servicios, sino también por el cambio que significa intentar recorrer y procesar este dominio de conocimientos e información tan nuevo y cambiante.

La aplicación de técnicas de WM, aunque apropiadas para este tipo de situaciones, no será tan sencilla como puede parecernos. Un lenguaje que procese esta BD tan excéntrica deberá adaptarse a las necesidades y características que hemos mencionado para la propia Web como BD, y por lo tanto deberá tener caracerísticas similares:

1. escalabilidad.
2. capacidad de abarcar cosas heterogéneas.
3. capacidad de compensar el crecimiento dinámico.

Veremos ahora que Java es el lenguaje indicado para realizar esta tarea y que su calidad de *open source* lo hace más potente, especialmente en la era en la que la mayoría de los especialistas en informática convenimos en la necesidad de promover el software abierto.

A.iii.Java como complemento del WM

En épocas pretéritas un lenguaje flexible, sencillo, transportable y escalable era algo difícil de pensar. Mucho más difícil era pensar que ese lenguaje fuera gratuito y altamente dinámico. Java cambió ese prejuicio y volvió realidad lo que parecía una utopía. Podrían mencionarse algunos problemas específicos que no hacen de Java una panacea, pero deben considerarse en la balanza las características de este lenguaje que lo vuelven único:

a - Es gratuito:

Pensar en un lenguaje tan extensamente usado en función de licencias, es cuestión de ciencia ficción. Si el ritmo de crecimiento de software continúa, la tendencia prioritaria hacia herramientas open source será mayor. Todo hace pensar que en un futuro cercano se cobrará por conocimientos e información y no por herramientas.

b - Es realmente transportable:

Al ser gratuito cualquiera, con cualquier plataforma y simplemente por su mera disposición bajará e instalará el kit de desarrollo. Esto hace que millones de personas estén permanentemente probando la transportabilidad de Java.

c - Es liviano:

A diferencia de muchas otras alternativas, con un diskette se puede tener un completo set de herramientas para desarrollar aplicaciones Java.

d - Crece permanentemente:

Existe una amplia comunidad muy activa que realiza mejoras al lenguaje. Estas mejoras se compilan y constituyen un major release cuando tienen suficiente jerarquía.

e - Mejora permanentemente:

Podría decirse que la comunidad Java es altamente crítica y por ello presiona constantemente en la calidad del lenguaje hasta el punto de que el lenguaje ha sido reescrito total o parcialmente cuando fue necesario para garantizar un diseño flexible, eficiente y transportable.

f - La comunidad es profesional:

Dado que el lenguaje se enraizó primeramente en la comunidad académica, el desarrollo y evolución del lenguaje respetó desde el principio ciertos paradigmas que trascienden la necesidad de cumplir un deadline para un release. Esto hoy en día permanece intacto y es la garantía esencial de la optimalidad del crecimiento y evolución del lenguaje Java.

g - Curva de aprendizaje razonable:

Puesto que es un lenguaje que cumple con metas ambiciosas, no es un lenguaje sencillo. A pesar de ello un programador relativamente entrenado tiene una curva de aprendizaje aceptable y en poco tiempo aumenta rápidamente su productividad.

h - Es flexible:

Se puede usar en sistemas tan heterogéneos como un microprocesador, un celular o una máquina vectorial.

i - Es potente:

La comunidad permamentemente desarrolla librerías para procesamientos complejos y potentes (paralelo, distribuido, clusters, etc.).

j - Es compatible con la Web:

Se necesita muy poco para colgar una aplicación Java de la Web. De hecho la concepción misma de Java responde a la necesidad de realizar aplicaciones Web. Java fue diseñado para vivir dentro y fuera de internet con muy pocos cambios.

k - Es un lenguaje joven:

Aunque parezca mentira es un lenguaje que nació en 1995 y en estos diez años ha tenido una evolución espectacular, como la evolución misma de la Web.

l - Es compacto:

Conocido el lenguaje, puede codificarse en manera clara y compacta. Una línea puede realizar muchas actividades a la vez (similar a lo que sucede con el lenguaje C).

m - Manejo localizado de errores:

El lenguaje resuelve por primera vez, de manera impecable el problema del manejo de los errores. Esto es esencial si se pretende implementar aplicaciones complejas.

n - Es escalable:

Una aplicación puede ser tan sencilla y eficiente como el programador lo desee. A medida que la habilidad del programador crezca sus programas serán más complejos, compactos, flexibles y eficientes de manera natural. Pero si el programador es novato, el lenguaje le facilitará sus primeras aplicaciones sin necesidad de que se convierta en experto desde el principio. Por otra parte, su naturaleza orientada a objetos permite el reuso de interfaces e implementación de manera bastante segura.

o - Concepto en capas:

Todo el lenguaje está ideado como un conjunto de capas de software que conviven. De esta forma los nuevos conceptos y los conceptos complejos se introducen naturalmente como librerías que tienen el mismo aspecto que el resto del lenguaje.

Respecto a las características de la Web pueden asociarse bastante directamente con las del lenguaje:

Tabla 1. Relación de las características de la Web con las de Java

Característica de la Web	*Característica de Java*
Gran tamaño	a, g, i, j, l, m, n
Contenido altamente heterogéneo	b,h, i, j, o
Gran dinamismo	c, d, e, f, g, h, j, k

Es bastante obvio que será Java el lenguaje elegido para nuestras implementaciones de WebMining.

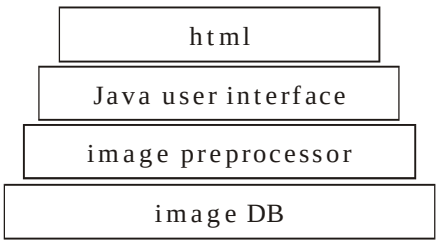
B. Aplicaciones

Vistos los conceptos generales, a continuación se presentarán brevemente un par de aplicaciones.

B.i.C-Bird

Esta es una aplicación desarrollada para hacer recuperación de información (o Information Retrieval, IR) a partir de una BD colgada de la Web. Su pecualiaridad es que realiza Web content mining sobre imágenes. Tiene una estructura en capas como se presenta en la Fig. 3.

Fig. 3. Estructura en capas de C-Bird

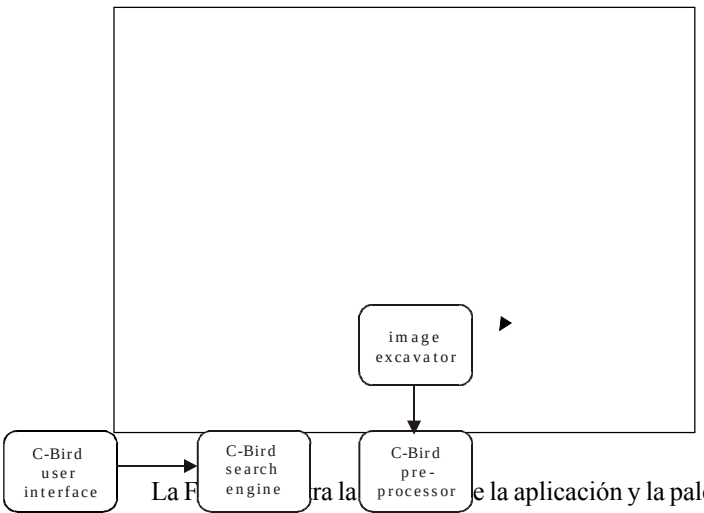


El funcionamiento de estas capas es bastante sencillo y obvio (Fig. 4). En un principio el *Image Excavator* recorre repositorios diversos con imágenes y/o texto (CD-Rom, discos rígidos, repositorios de video, etc.) y extrae las imágenes (y video) para pasarlas al *Pre-processor*. El *Pre-processor* a su vez extrae características de las imágenes (tiene una cantidad variada de algoritmos previendo las distintas formas de procesar o buscar una imagen) y les asocia texto y metadatos (información acerca del significado del contenido). Luego las almacena. Una consulta se introduce por una interface especial

(*User Interface*) que la preprocesa y extrae sus características para que el *Search Engine* pueda compararlas contra las que aparecen almacenadas en la BD. Cuando se produce una coincidencia entre ambas cosas se recuperan las imágenes relacionadas.

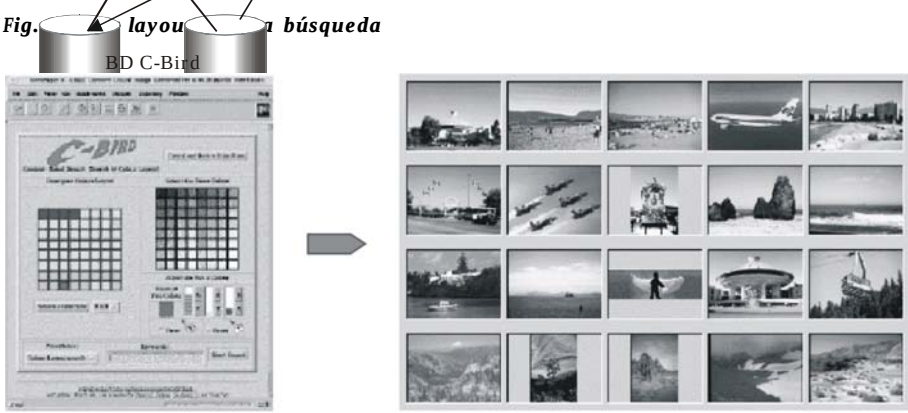
A manera de ejemplo imaginemos que deseamos realizar una búsqueda. Se pueden emplear distintos criterios ya conocidos por la aplicación (histogramas de colores, color layout, cromaticidad, etc.). Supongamos que se desea buscar por un color layout de 8 x 8. Cuando se realiza este tipo de búsqueda, se define la paleta de colores deseada. Se puede trabajar con paletas de 1x1, 2x2, 3x3, 4x4 u 8x8 colores. En nuestro caso trabajaremos sobre una de 8x8.

Fig. 4. Secuencia de pasos en C-Bird



La F[illegible]ra la [illegible]e la aplicación y la paleta seleccionada. A su derecha se muestra el resultado obtenido con este criterio.

Fig. [illegible] layou[illegible]a búsqueda



Esta *respuesta*, al menos en principio, parece un poco *extensa*. Si se pretende mayor precisión en la búsqueda se puede optar en esta herramienta por utilizar como complemento los keywords o metadatos que el sistema ya tiene asociados a todas y cada una de las imágenes, por ejemplo el nombre de la región fotografiada y la palabra montaña. El resultado en nuestro caso es asombrosamente más preciso. Como se ve en la Fig. 6, la respuesta se reduce a un único resultado.

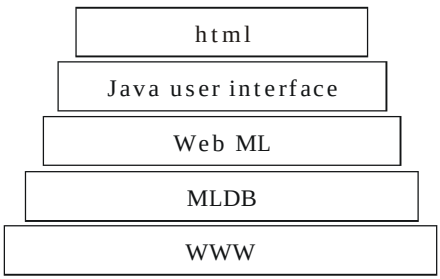
Fig. 6. Resultado con apoyo de metadatos

B.ii.VWV

Otro caso es el Virtual Web View (VWV), un sistema pensado para realizar otro tipo de Web Mining como complemento a una actividad de navegación ya no sobre páginas sino sobre una estructura que represente la información contenida (Information Browsing). También fue diseñado para tareas de descubrimiento de recursos o información tácita (Resource Discovering).

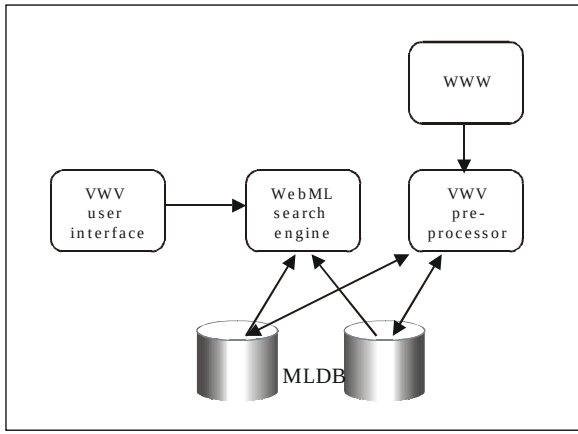
Esta aplicación, como es de esperar, también tiene una estructura en capas (ver Fig. 7) donde se destacan los distintos componentes que veremos funcionando a continuación (Fig. 8).

Fig. 7. Estructura en capas de VWV



Inicialmente el sistema procesa toda la porción de WWW definidas dentro del alcance de la aplicación.

Fig. 8. Funcionamiento de VWV



Como parte de este procesamiento el *preprocessor* extrae la información textual y metadatos correspondientes a los objetos no textuales de la Web. Esta información la almacena en una Multi-Layer Data Base (MLDB), una BD especial donde los datos se organizan en jerarquías ontológicas predefinidas. Cuando se recibe una consulta, ésta se captura con la *user interface* del sistema, quien realiza un *parsing* adecuado que permite reducir la consulta a un formato compatible con WebML. El WebML es un lenguage del tipo *Markup Language* (similar al XML) que permite realizar comparaciones con el contenido de la MLDB. Los resultados de esta búsqueda se presentarán al usuario como respuesta a la consulta realizada.

Bibliografía

[1] S. Abitibout. Querying semi-structured data. In Int. Conf. on Database Theory, 1997.

[2] R. Agrawal and J. C. Shafer. Parallel mining of association rules: Design, implementation, and experience. IEEE Trans. Knowledge and Data Engineering, 8:962/969,1996.

[3] AI Magazine, 18(2). Intelligent Systems on the Internet, 1997.

[4] AI Magazine, 19(2). Intelligent Agents, 1998.

[5] American National Standards Institute. Database Language SQL, ansi x3.135-1992 edition, 1992.

[6] Yigal Arens, Chun-Nan Hsu, and Craig A. Knoblock. Query processing in the sims information mediator. In ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop, 1996.

[7] Roberts Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. WebWatcher: A learning apprentice for the world wide web. In AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, March 1995.

[8] J.R. Bach, C. Fuller, A. Gupta, and et al. The Virage image search engine: An open framework for image management. In SPIE Storage and Retrieval for Image and Video Databases IV, February 1996.

[9] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111/122, 1981.

[10] D.H. Ballard and C.M. Brown. Computer Vision. Prentice Hall, 1982.

[11] P.M.E. De Bra and R.D.J. Post. Information retrieval in the world-wide web: Making client-based searching feasible.

[12] Jeffrey M. Bradshaw. Software Agents. AAAI Press / The MIT Press, 1997.

[13] P.J. Burt. Smart sensing within a pyramid vision machine. Proceedings of IEEE, 76(8):1006/1015, 1988.

[14] D.W. Cheung, V. Ng, A. Fu, and Y. Fu. E_cient mining of association rules in distributed databases. IEEE Trans. Knowledge and Data Engineering, 8:911/922, 1996.

[15] Je_ Conklin. Hypertext: An introduction and survey. IEEE ComputerDatabase Engineering, 20(9):17/41, September 1987.

[16] E.H. Durfee, D.L. Kiskis, and W.P. Birmingham. The agent architecture of the University of Michigan digital library. IEEE Software Engineering, 144(1):61 {71, February 1997.

[17] Mark T. Maybury Editor. Intelligent Multimedia Information Retrieval. The AAAI Press/The MIT Press, 1997.

[18] Max J. Egenhofer. Spatial Query Languages. PhD thesis, University of Maine, 1989.

[19] Andrew Fall. Reasoning with Taxonomies. PhD thesis, School of Computing Science, Simon Fraser University, December 1996.

[20] U. M. Fayyad, S. G. Djorgovski, and N. Weir. Automating the analysis and cataloging of sky surveys. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 471/493. AAAI/MIT Press, 1996.

[21] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.). Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[22] M. Flickner, H. Sawhney, W. Niblack, and et al. Query by image and video content: The QBIC system. IEEE Computer, 28(9):23/32, September 1995.

[23] Yiangjian Fu. Discovery of Multiple-level Rules from Large Databases. PhDthesis, School of Computing Science, Simon Fraser University, July 1996.

[24] B.V. Funt and G.D. Finlayson. Color constant color indexing. IEEE Trans. Patt.Anal. andMach. Intell., 17:522/529, 1995.

[25] Athula Ginige, David B. Lowe, and John Robertson. Hypermedia authoring. IEEE Multimedia, pages 24/34, 1995.

[26] V. Gudivada and V. Raghavan. Content-based image retrieval systems. IEEE Computer, 28(9):18/22, 1995.

[27] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowledge and Data Engineering, 5:29/40, 1993.

[28] T.H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-6(2):222/229, 1984.

[29] S. Khosha_an and A. B. Baker. Multimedia and Imaging Databases. Morgan Kaufmann Publishers, 1996.

[30] Krzysztof Koperski. A Progressive Re_nement Approach for Spatial Data Mining. PhD thesis, School of Computing Science, Simon Fraser University, April 1999.

[31] H. F. Korth and A. Silberschatz. Database System Concepts, 2ed. McGraw-Hill, 1991.

[32] Z.N. Li and B. Yan. Recognition kernel for content-based search. In Proc. IEEE Conf. on Systems, Man, and Cybernetics, pages 472/477, 1996.

[33] F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(7):722/733, 1996.

[34] W. Lu, J. Han, and B. C. Ooi. Knowledge discovery in large spatial databases. In Far East Workshop on Geographic Information Systems, pages 275/289, Singapore, June 1993.

[35] M. Antonini, et al. Image coding using wavelet transform. IEEE Trans. on Image Processing, 1(2):205/221, 1992.

[36] J. Ostermann, E.S. Jang, J. Shin, and T. Chen. Coding of arbitrarily shaped video objects in MPEG-4. In Proc. Int. Conf. on Image Processing (ICIP '97), pages 496/499, 1997.

[37] G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/ MIT Press, 1991.

[38] R.M. Haralick, et al. Texture features for image classiffication. IEEE Trans. on Systems, Man, and Cybernetics, SMC-3(6):610/621, 1973.

[39] G. Salton. An automatic phrase matching. In D. Hays, editor, Readings in Automatic Language Processing. American Elsevier Publishing Company Inc., New York, 1966.

[40] R.M. Haralick, et al. Texture features for image classification. IEEE Trans. on Systems, Man, and Cybernetics, SMC-3(6):610/621, 1973.

[41] C.S. Roberts. Partial-match retrieval via the method of superimposed codes. Proc. IEEE, 67(12):1624/164, December 1979.

[42] G. Salton. An automatic phrase matching. In D. Hays, editor, Readings in Automatic Language Processing. American Elsevier Publishing Company Inc., New York, 1966.

[43] J.R. Smith and S.F. Chang. Visually searching the web for content. IEEE Multimedia,4(3):12/20, 1997.

[44] T.R. Smith. A digital library for geographically referenced materials. IEEE Computer, 29(5):54/60, 1996.

[45] P. Stolorz, H. Nakamura, E. Mesrobian, R.R. Muntz, E.C. Shek, J.R. Santos, J. Yi, K. Ng, S.Y Chien, C.R. Mechoso, and J.D. Farrara. Fast spatio-temporal data mining of large geophysical datasets. In Proc. First Int. Conf. On Knowledge Discovery and Data Mining, pages 300/305, August 1995.

[46] Ronald J. Vetter, Chris Spell, and Charles Ward. Mosaic and the world-wide web. IEEE Computer, 27(10):49/57, October 1994.

[47] Jie Wei. Foveate Wavelet Transform and its Applications in Digital Video Processing, Acquisition, and Indexing. PhD thesis, School of Computing Science, Simon Fraser University, November 1998.

[48] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin core metadata for resource discovery. Request for Comments rfc2413, September 1998.

[49] W.A. Woods. Important issues in knowledge representation. Proc. of the IEEE, 74(10), October 1986.

[50] Osmar R. Zaïane and Jiawei Han. Webml: Querying the world-wide web for resourcesand knowledge. In Proc. ACM CIKM'98 Workshop on Web Information and Data Management (WIDM'98), pages 9/12, Washington DC, November 1998.

[51] Hua Zhu. On-line analytical mining of association rules. Master's thesis, School of Computing Science, Simon Fraser University, December 1998.