Proyecto 7: un motor de recuperación de información web colaborativo

Proyecto 7: a collaborative information retrieval engine

Carlos García FIGUEROLA PANIAGUA, Raquel GÓMEZ DÍAZ,
José Luis ALONSO BERROCAL, José Francisco ZAZO RODRÍGUEZ

Facultad de Traducción y Documentación. Universidad de Salamanca. C/ Fco. Vitoria, 6-16; 37008 (Salamanca), {figue,rgomez,berrocal,zazo}@usal.es

Resumen

Se presenta el Proyecto 7, un motor de recuperación web pensado para su aplicación por parte de las organizaciones con presencia documental en Internet y también para las que, aún teniendo la información almacenada en modo exclusivamente local, quieren hacerla accesible a través de Internet. Se repasan los fundamentos teóricos en los que se apoya este motor de recuperación web y se exponen sus principales características.

Palabras clave: Recuperación de información. Motor de búsqueda. Recuperación web. Proyecto 7. Web crawling. Modelo probabilístico.

Abstract

The Project 7 is presented, a web retrieval engine designed for use by those organizations with a documentary presence in Internet; but also for those that, having stored their infomation in a exclusively local way, want to make it accessible via Internet. The theoretical basis on which this web retrieval engine is based is reviewed, outlining its main features.

Keywords: Information retrieval. Search engine. Web retrieval. Project 7. Web crawling. Probalistic retrieval.

1. Introducción

Con la expansión de Internet como medio de difusión de información se hace patente la necesidad de contar con medios de recuperación que permitan acceder a la información deseada de manera fácil y rápida. Los motores de tipo generalista (Yahoo, Google, Bing, etc.) e incluso los especializados temáticamente pretenden, con mayor o menor fortuna, cubrir todo el espacio web (Pulido, 1997). Este tipo de motores de búsqueda han demostrado su eficacia, y la prueba es que son utilizados a diario por millones de personas.

En ocasiones, sin embargo, las organizaciones con presencia en la red desean disponer de herramientas de búsqueda dentro de su propio espacio de red, ya sea éste abierto al público, solamente interno, o una mezcla de ambas cosas. Lo que con frecuencia muchas de estas organizaciones pretenden es disponer de motores de búsqueda centrados en su espacio web (Broder, 2002), que la propia organización pueda controlar con precisión, permitiendo un mayor control sobre el espacio web indizado, sobre las formas de acceso, la presentación de la información recuperada e, incluso, la posibilidad de estudiar hábitos de búsqueda a fin de de organizar dicho espacio web y su navegabilidad de una forma más eficiente (Chi, Pirolli, Chen y otros, 2001).

En este trabajo presentamos el Proyecto 7, un motor de recuperación web pensado para su aplicación por parte de esas organizaciones con presencia documental en Internet y también para las que, aún teniendo la información almacenada en modo exclusivamente local, quieren hacerla accesible a través de Internet.

Este trabajo está organizado como sigue: en primer lugar se hace un repaso a las soluciones de indización y búsqueda existentes; en la sección siguiente se exponen las bases teóricas en las que se asienta el motor de recuperación diseñado. A continuación se expone la génesis del Proyecto 7 y de su motor de búsqueda, para detallar después las características técnicas de dicho motor, así como la forma en que ese software se distribuye. Finalmente, se ofrecen unas conclusiones y unas líneas de trabajo futuro.

2. Instrumentos de recuperación

Como se ha comentado antes, en ocasiones las organizaciones necesitan disponer de motores de búsqueda que indicen sólo una parte del espacio web, por lo general el dominio perteneciente a la organización en cuestión. Una alternativa obvia son los propios sistemas de bús-

queda que forman parte de muchos gestores de contenidos web o CMS (Content System Management); sin embargo, muchos de tales sistemas incorporados en los CMS son extremadamente simples y carecen de prestaciones que hoy consideramos básicas, como, por ejemplo, una gestión razonable de acentos y otros diacríticos (Wikipedia, 2010).

En cualquier caso, una limitación importante es que su ámbito de búsqueda se circunscribe a los documentos gestionados por el propio CMS. Pero muchas veces los dominios de una organización son algo más complejo que un simple portal web: diversos servidores con contenidos y software para gestionarlo heterogéneo, que están fuera del ámbito de actuación de un CMS concreto. Se impone en estos casos el uso de un motor de dominio como herramienta más adecuada.

2.1. Motores de recuperación de dominios

Podemos entender como 'motor de recuperación' de dominios web, de manera informal, un software que es capaz de explorar automáticamente un dominio web (o la parte de él que se desee), recopilando e indizando las páginas y otro tipo de recursos que pueda encontrar en esa exploración; permitiendo búsquedas posteriores, naturalmente. El interfaz de búsqueda es a través del propio web, naturalmente.

Así, un motor de recuperación de este tipo consta de tres componentes básicos: un crawler que explora la red y recopila documentos, un indizador que analiza esos documentos que obtiene el crawler y los indiza; y un módulo de búsqueda, que permite hacer búsquedas a partir de esos índices.

Lo que marca la diferencia más importante con otro tipo de sistemas de recuperación es el crawler (Girardi, 2006; Olston, 2010), pero no es la única. A diferencia de otros sistemas documentales más o menos estáticos, el web está en cambio permanente: continuamente aparecen documentos nuevos, pero otros que ya estaban se modifican o desaparecen (Kaunder, 2010); los módulos de indización necesitan ser actualizados permanentemente.

También, y en contra de lo que podría pensarse a primera vista, los formatos de los documentos son heterogéneos: no sólo hay HTML en los dominios web, sino que realmente nos encontramos con recursos de lo más variado, y el software de los motores de recuperación tiene también que enfrentarse a ello.

Diversos programas se han diseñado para afrontar estos cometidos. No pretendemos hacer aquí una revisión exhaustiva, sino simplemente apuntar algunos datos sobre los paquetes más conocidos. En este sentido, hay paquetes completos, que contienen todo lo necesario para disponer de un sistema de recuperación de dominio; los hay que deben ser completados con terceros programas en alguna de sus funciones básicas; y existen componentes o herramientas de programación específicas que permiten diseñar y construir motores de recuperación de dominios web (Beigbeder, 2006).

Entre los primeros, algunos de los más venerables, por su antigüedad, podrían ser WebGlimpse (Manber, 1997; Internet Workshop, 2010), mngoSearch (LavTech.com Corporation, 2010) o htdig (The ht://dig Group, 2005; Rato, 2003), este último sin mantenimiento efectivo desde 2004, pero que goza todavía de una gran popularidad.

Más recientes son productos como Nutch (The Apache Software Foundation, 2010a; Cafarella y Etzioni, 2005) o Solr (The Apache Software Foundation, 2010b; Smiley y Pugh, 2009), ambos apadrinados por Apache, la fundación que mantiene el superconocido servidor web. Ambos están escritos en Java y están pensados para ser utilizados a través de Tomcat, de manera que su administración es todo menos sencilla. En este mismo grupo podemos situar un producto poco difundido, pero potente llamado Datapark.

El propio Google tiene también su producto comercial en este campo: un motor de recuperación para sitios de diversos tamaño (Google Inc., 2010).

Entre los segundos, uno de los más conocidos es Swish-E (Swish-e, 2010; Rabinovitz, 2003). Éste es un potente motor de recuperación que incluye características interesantes, como la posibilidad de pesar diferenciadamente campos XML y un sinfín de parámetros ajustables mediante directivas a través de un fichero de configuración. Está preparado para recibir documentos de un crawler, e incluso tiene su propio crawler, pero es muy limitado y la propia documentación oficial del producto recomienda utilizar cualquier otro.

Terrier (University of Glasgow, Terrier Team, 2010; Ounis, Amatis, Plachouras y Otros, 2005), basado en Java, carece de crawler, aunque recomienda utilizar para tales funciones Labrador (http://www.dcs.gla.ac.uk/~craigm/labrador/index.html), un crawler autónomo diseñado precisa-mente para complementar a Terrier.

En parecida línea, Lemur (The Lemur Project, 2009; Allan, Callan Collins-Thompson y Otros, 2003), aunque diseñado para tareas de investi-

gación y experimentación en el campo de la Recuperación de Información, goza en los últimos tiempos de gran difusión. Carece, sin embargo, de crawler, por lo que debe ser utilizado con un componente externo de este tipo.

Finalmente, Omega (Beigbeder, Buntine y Yee, 2006), un motor de recuperación probabilístico, pero que también carece de crawler. Tiene, no obstante, diversas utilidades que permiten importar las colecciones de documentos recolectadas por otros crawlers y sistemas, como HtDig.

Sobre componentes y herramientas de programación específicas para construir motores de recuperación, algunas se han citado ya. En primer lugar, sistemas como Terrier o Lemur, que, además de ser en sí mismos motores de recuperación completamente desarrollados, permiten acceder a sus componentes desde diversos lenguajes de programación, de manera que es relativamente fácil, a partir de ellos, diseñar nuevos motores de recuperación.

La conocida librería Lucene (Paz-Madrid y Otros, 2007), de Apache Foundation, es la base de los mencionados sistemas Nutch y Solr; tiene *ports* o *bindings* para diferentes lenguajes de programación, y es la base de muchos sistemas de recuperación plenamente operativos. También es utilizada internamente por varios Gestores de Contenidos o CMS, como el conocido D-space, un software gestor de repositorios digitales.

Xapian (The Xapian Project, 2010) es una librería basada en el modelo probabilístico, con la cual se ha diseñado el sistema de recuperación Omega, citado antes. Cheshire es otro conjunto de herramientas interesante (http://www.cheshire3.org/), aunque pensado inicialmente para sistemas documentales locales. Sus capacidades para manejarse con protocolos como Z.3950 o OAI hacen que sea una alternativa interesante.

3. La recuperación probabilística

La librería Xapian, citada más arriba, es el instrumento o componente esencial en el motor que presentamos en este trabajo. Tiene la particularidad de estar basada en el llamado modelo probabilístico de recuperación (Vilares, 2008). Éste es un modelo teórico, pero que tiene repercusiones de tipo práctico, y que no es frecuente encontrar en otros motores.

El llamado modelo probabilístico se basa en los trabajos de Karen Sparck Jones y Stephen Robertson (Jones y Robertson, 1998), por citar dos de los nombres más significativos, y ha demostrado a lo largo de sus 30 años su eficacia. Básicamente, considera el proceso de recuperación como un proceso de aproximación al conjunto teórico de documentos relevantes para una consulta dada.

Esta aproximación se realiza calculando la probabilidad de que cada documento pertenezca a dicho conjunto de relevantes. Como dicha probabilidad no es conocida, hay que calcularla; para el modelo probabilístico los documentos están definidos por los términos que contienen, por lo que la probabilidad de que un documento sea relevante podría ser estimada a través de la probabilidad de que cada término que contiene índice un documento relevante.

Como dicha probabilidad de los términos tampoco es conocida, puede aplicarse una probabilidad inicial (basada, por ejemplo, en frecuencias de aparición en el documento y en la colección y en la coincidencia con los términos de la consulta). Esta probabilidad inicial puede ser refinada posteriormente a través de un proceso iterativo, hasta estimar una probabilidad razonable.

Xapian se basa en estas ideas, y calcula pesos para los términos de los documentos aplicando el esquema conocido como BM25 (Robertson, 2004), que ha sido exitosamente utilizado en sistemas operacionales y experimentales durante bastante tiempo.

4. El proyecto 7

En este contexto se plantea lo que hemos llamado Proyecto 7, un motor de búsqueda para dominios web plenamente operativo, que surge de la confluencia entre la disponibilidad de herramientas informáticas adecuadas por su potencia y sencillez; y de las actividades docentes en torno a una nueva asignatura en el Grado en Documentación.

Esta asignatura, titulada Recuperación Automatizada de la Información, pretende impartirse como una especie de aplicación práctica de los conceptos impartidos en otra asignatura anterior en el tiempo: Técnicas de Indización y Recuperación de la Información. Y, en efecto, ¿qué mejor aplicación práctica de dichos conceptos teóricos, que la construcción y puesta en explotación de un motor de recuperación de este tipo?

En la actualidad, el motor 7 está plenamente operativo, es decir, se encuentra en funcionamiento, aunque obviamente es susceptible de mejoras, ampliaciones, etc. Un cuadro resumen de sus funcionalidades puede verse en la Tabla I.

Las características más importantes del motor 7 pueden exponerse a través de los grandes bloques funcionales de cualquier sistema de este

tipo. De forma genérica, el motor 7 es un software basada en la librería Xapian, más concretamente en su adaptación a Python. Es un conjunto de dos programas, ambos muy ligeros, como también lo es el propio lenguaje Python, interpretado, con que están escritos.

Crawling	*Interfaz gráfica*	*En desarrollo*
	Limitación de tiempo, número de documento, tamaño de documentos	Configurable
	Determinación de formato de documentos	A través de Magic BSD FF (no por la extensión de los ficheros)
	Filtrado de documentos por formato	Sí
	Conversión a texto plano para indización	Sí, mediante conversores externos (pdt2text, etc.), configurable
	Filtrado de URLs a explorar	Configurable, por medio de reglas basadas en expresiones regulares
	Límites especifícicos para servidores determinados	Sí
	Multihilo	En desarrollo
Indización	Modelo teórico	probabilístico
	Pesado de términos	Okapi BM25
Análisis léxico	stemmimg	Opcional, dependiente del idioma
	Eliminación de palabras vacías	Opcional
	Corrección ortotipográfica	Sí
Búsquedas	Interfaz web	Sí
	Búsquedas simples	Sí
	Sugerencia de términos	En desarrollo
	Operadores booleanos	Sí (AND, OR, NOT)
	Operadores de adyacencia	Sí (NEAR, ADJ)
	Búsqueda por frases	Sí
	Búsqueda por facetas	En desarrollo
Resultados	Interfaz web	Sí
	Paginación Configurable	Sí
	Ordenación de documentos por relevancia	Sí
	Criterio de ranking	Similitud probabilística con la consulta. Implementación de otros criterios en desarrollo
	Snnippet (resumen) de documento	Sí, 800 caracteres
	Enlace activo a documento	Sí
	Eliminación de documentos duplicados	Sí, mediante MD5. Duplicados aproximados en desarrollo

Tabla I. Resumen de características

El código del módulo buscador, por ejemplo, no supera las 200 líneas; el indizador es algo más extenso, al incorporar un crawler, pero se mantiene también dentro de esa línea, por lo que ambos son fáciles de analizar, comprender y modificar por un programador.

Tanto Python como las librerías y módulos que utiliza son portables entre sistemas operativos, por lo que corre sin problemas sin necesidad de adaptaciones en cualquier tipo de máquina.

Consta, básicamente, de dos *scripts* o programas interpretados, más algún otro programa de utilidad, además de una hoja de estilo para con-

trolar la visualización y formato de la interfaz web del motor. Se utiliza junto con el servidor web Apache, y requiere que éste cargue el módulo estándar para Python (mod-python); el código Python, por otra parte, sólo utiliza módulos estándar del lenguaje, aparte de Xapian, naturalmente.

4.1. Crawling

El crawler que incorpora es el módulo encargado de explorar de forma automática el web, recopilando páginas y documentos en general que habrán de ser indizados. Controla la porción de espacio web que ha de recorrer mediante reglas basadas en expresiones regulares y está preparado para sortear las conocidas 'trampas para robots' que suelen aparecer en este tipo de exploraciones (Figuerola, Berrocal, Zazo y Otros, 2006). Incorpora límites ajustables de seguridad, de tiempo, número de documentos y tamaño; y respeta el 'Estándar de Exclusión de Robots' (Koster, 2004). En la misma línea de exploración 'educada', hace una pausa, también configurable, entre peticiones al mismo servidor.

El crawler es capaz de gestionar adecuadamente los códigos de error de los servidores web, así como las redirecciones; los errores de timeout se colocan en una cola de reintentos, cuyo número es configurable; la gestión de los reintentos de error por timeout, junto con los límites de tamaño de documento permiten trabajar con segmentos de red de escaso ancho de banda, si es preciso.

El crawler actual es monohilo, a pesar de lo cual su velocidad de exploración es más que suficiente para recorrer o explorar dominios de tamaño mediano en muy poco tiempo. Naturalmente, la velocidad de exploración depende de factores externos como el ancho de banda y la rapidez de respuesta de los servidores explorados (también de la duración que hayamos configurado en nuestro robot para la pausa entre petición y petición); a título indicativo, un dominio de unas 150.000 páginas puede ser recorrido e indizado en unas 12 horas.

4.2. Indización

El crawler incorpora el módulo de indización, por lo que las páginas que el crawler explora son pasadas automáticamente al módulo indizador. Éste no sólo indiza páginas web, sino todo tipo de documentos; la condición es que éstos puedan ser convertidos a texto plano. Esta conversión se efectúa mediante conversores externos, que han de ser llamados de manera automática por el módulo indizador. Existen conversores de este tipo para multitud de formatos de documentos electrónicos; ciñéndonos sólo a los que son libres, de código abierto y con versiones para Windows, Linux y Mac, tenemos conversores para PDF, Postscript, MS Word, Excel y PowerPoint, RTF, Open Writer, Open Calc, Open Impress, Koffice, etc.

El indizador es capaz de manejar adecuadamente cualquier tipo de código de caracteres, aun-que internamente trabaja con UTF-8, actual estándar para HTML-XML; sin embargo, otros juegos de caracteres menos evolucionados pero todavía frecuentes (ISO-8859, etc.) son manejados sin problemas. Esto es especialmente importante cuando se quiere trabajar con documentos que están en ordenadores heterogéneos, con sistemas operativos y software muy diverso. Dentro de las características del análisis léxico, se pueden configurar listas de palabras vacías, sinónimos y, parcialmente por el momento, acrónimos.

El crawler-indizador, que no es accesible, obviamente, a los usuarios, se adapta a las condiciones de funcionamiento deseadas mediante un fichero de configuración. Por el momento, su interfaz es en modo texto.

4.3. Búsquedas

Las búsquedas se efectúan median un interfaz web, cuyo formulario admite búsquedas mediante lenguaje natural, y también el uso de operadores. Éstos pueden ser los clásicos booleanos, pero también operadores de adyacencia; el sistema admite búsquedas de frases, así como una mezcla o combinación de todo lo anterior.

Los resultados se ofrecen mediante una o varias páginas web, ordenados por relevancia. Para cada documento encontrado, además de su enlace correspondiente y otros elementos de información (título, formato, un resumen o snnipet), se ofrece la posibilidad de ser utilizado como modelo para expandir la consulta, recuperando nuevos documentos similares a ése (Zazo, Berrocal, Figuerola y otros, 2005).

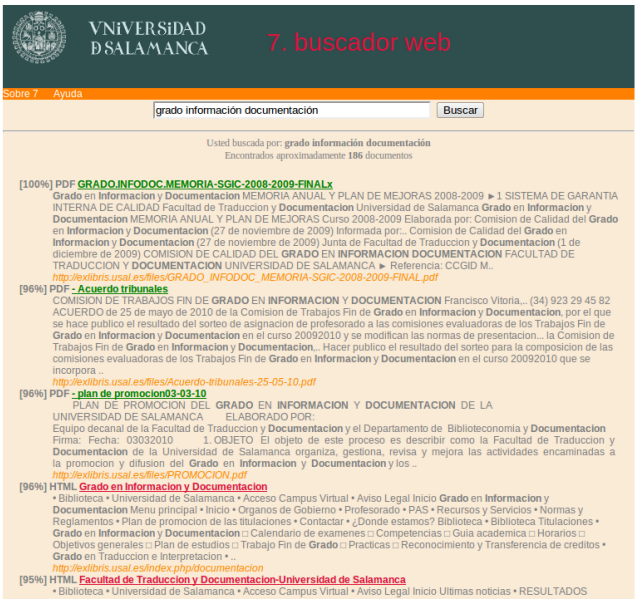

Figura 1. Resultados de una búsqueda en el dominio de la Universidad de Salamanca (aprox. 100 000 documentos web, 57 servidores diferentes)

De manera opcional, el sistema puede almacenar internamente las consultas efectuadas y los enlaces o documentos, de entre los recuperados, navegados después. Estos datos, junto con los recogidos en el log del propio servidor web, puede ser muy útiles para estudiar las pautas y comportamientos de los usuarios (Zazo, Figuerola, Berrocal y otros, 2004).

4.4. Distribución

El motor de recuperación es software libre, y se distribuye como tal. En consecuencia, el código es abierto, puede ser estudiado, modificado, mejorado y completado o complementado con nuevas funcionalidades por quien desee.

No podía ser de otra forma, puesto que software libre son los distintos componentes que se han utilizado en su diseño. En consecuencia, puede ser descargado libre y gratuitamente del su sitio web, junto con instrucciones detalladas de instalación, tanto para Linux como para Windows.

5. Trabajo futuro

Diversas líneas de trabajo están planteadas, siempre sin perder de vista que éste es, en gran medida, un proyecto vinculado a la docencia en Recuperación de Información. A grandes rasgos, lo que se pretende es que los propios estudiantes del Grado en Documentación detecten necesidades a implementar, mejoras aconsejables y errores o fallos a eliminar.

Paralelamente, está previsto incluir en esta dinámica a estudiantes con un perfil más tecnológico, como los participantes en algún postgrado en informática, que sean quienes aborden los aspectos más técnicos (programación, codificación, etc.) de tales mejoras o correcciones.

En cualquier caso, algunas mejoras están ya planteadas: por ejemplo, una gestión eficaz de acrónimos en indización y búsqueda, que está a punto de conseguirse en el momento de escribir este trabajo.

También, aplicación de clustering jerárquico en la visualización de resultados, unas técnicas que ya hemos utilizado en otros trabajos (Mateos Sánchez y Figuerola, 2009), y que se han mostrado muy eficaces cuando las consultas pueden tener un grado alto de ambigüedad. Más sencillo, desde el punto de vista tecnológico, pero algo que otros sistemas (Google Enterprise, por ejemplo) utilizan es el agrupamiento de resultados por servidores de origen o URLs; nuestro objetivo es implementarlo, de forma opcional o configurable, aunque no está clara la utilidad en la recuperación, desde el punto de vista del usuario.

Más urgente parece mejorar la calidad de los snnipets de los documentos recuperados; éste es un elemento que parece importante para que el usuario decida o no acceder al documento, y tal vez proseguir la navegación a partir de él. Técnicamente existen diversas posibilidades, pero éstas (su utilidad final) han de ser probadas y evaluadas debidamente.

6. Conclusiones

Hemos presentado el Proyecto 7, un motor de recuperación de dominios web surgido de la confluencia entre la docencia en materias de Recuperación de Información, la disponibilidad de instrumentos y la experiencia de nuestro grupo en este tipo de temas. Se han descrito las características más importantes de dicho motor de recuperación y se han marcado líneas de trabajo futuro en la línea de mejorar y añadir nuevas prestaciones a ese sistema de recuperación.

Referencias

Allan, J.; Callan, J.; Collins-Thompson, K. y Otros (2003). The lemur toolkit for language modeling and information retrieval. (2008-01-03).

Beigbeder, M.; Buntine, W.; Yee, W.G. (2006). Open source search and research. // Proceedings of the 2006 international workshop on Research issues in digital libraries. (2006) 5 y ss.

Broder, A. (2002). A taxonomy of web search. // ACM Sigir Forum. 36:2 (2002) 3-10.

Cafarella, Michael J.; Etzioni, Oren (2005). A Search Engine for Natural Language Applications. // Proceedings of the 14th International World Wide Web Conference (WWW 2005).

Chi, E.H.; Pirolli, p. ; Chen, K.; Pitkow, J. (2001). Using information scent to model user information needs and actions and the Web. // Proceedings of the SIGCHI conference on Human factors in computing systems. (2001) 497 y ss.

Figuerola, Carlos G.; Alonso Berrocal, José Luis; Zazo Rodríguez, Ángel F.; Rodríguez Vázquez de Aldana, Emilio (2006). Diseño de Spiders. Departamento de Informática y Automática - Universidad de Salamanca. Informe Técnico DPTOIA-IT-2006-002 (2006). http://reina.usal.es/papers/figuerola2006diseno.pdf (2010-04-01).

Girardi, C.; Ricca, F.; Tonella, p. (2006). Web crawlers compared. // International Journal of Web Information Systems. 2:2 (2006) 85-94.

Google Inc. (2010). Tecnología de Google para la empresa. http://www.google.es/enterprise (2010-04-01).

Internet Workshop (2010). WebGlimpse and Glimpse: advanced site search software for Unix: index web-sites or intranets. http://webglimpse.net/ (2010-04-01).

Jones, K.S.; Walker, S.; Robertson, S.E.(1998). A probabilistic model of information retrieval: development and status. // Information Processing and Management. 36:6 (1998) 809-840.

Kaunder, Maurice de (2010). WorlWideWebsize.com: The size of the World Wide Web. http://wolrdwidesize.com (2010-04-01).

Koster, M. (1994). A standard for robot exclusion, 1994. http://info.webcrawler.com/mak/projects/robots/norobots.html (2010-04-01).

LavTech.com Corp. (2010). mnoGoSearch - Internet Search Engine Software. http://www.mnogosearch.net (2010-04-01).

Manber, U.; Smith, M.; Gopal, B. (1997). Webglimpse: Combining browsing and searching. // Proceedings of the annual conference on USENIX Annual Technical Conference (1997) 15 y ss.

Mateos Sánchez, Montserrat; G. Figuerola, Carlos (2009). Aplicación de técnicas de clustering en la recuperación de información web. Gijón: Ediciones TREA, 2009.

Olston, C.; Najork, M. (2010). Web Crawling. // Information Retrieval. 4:3 (2010) 175-246 .

Ounis, I.; Amati, G.; Plachouras V.; He, B.; Macdonald, C.; Johnson, J. (2005). Terrier Information Retrieval Platform. // Proceedings of the 27th European Conference on IR Research (ECIR 2005). Lecture Notes in Computer Science. 3408 (2005) 517-519.

Paz-Madrid Gorelov, Vadim; Zazo, Ángel F.; Figuerola, Carlos G.; Alonso Berrocal, José Luis (2007). Librerías Lucene y dotLucene para Recuperación de Información. Estudio y desarrollo de casos prácticos. Departamento de Informática y Automática - Universidad de Salamanca. Informe Técnico DPTOIA-IT-2007-003 (2007) http://reina.usal.es/papers/pazmadrid2007librerias.pdf.

Pulido, J.R.V. (1997). Recuperación de la información en Internet: motores y otros agentes de búsqueda. // Scire. 3:2 (1997) 85 y ss.

Rabinovitz, Josh (2003). How to Index Anything. // Linux Journal. July 2003, 82-88.

Rato González, C. (2003). HTDIG, el detective en la red. // Solo Programadores Linux. 53 (2003) 34-38.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. // Journal of Documentation. 60 (2004) 503-520.

Smiley, David; Pugh, Eric (2009). Solr 1.4. Enterprise Search Server. Birmingham UK: Packt Publishing, 2009.

Swish-e (2007). Simple Web Indexing System for Humans - Enhanced. http://swish-e.org (2010-04-01).

The Apache Software Foundation (2010). Welcome to Nutch. http://http://lucene.apache.org/nutch (2010-04-01)

The Apache Software Foundation (2010). Welcome to Solr. http://lucene.apache.org/solr/ (2010-04-01).

The ht://Dig Group (2007). WWW Seasrch Engine Software. http://www.htdig.org (2010-04-01).

The Lemur Project (2009). The Lemur Toolkit for Language Modeling and Information Retrieval. http://www.lemurproject.org/ (2010-04-01).

University of Glasgow, Terrier Team (2010). Terrier IR Platform. http://terrier.org (2010-04-01).

Vilares, J. (2008). El modelo probabilístico: características y modelos derivados. // Revista General de Información y Documentación. 18 (2008) 343-363.

Wikipedia (2010). Sistema de gestión de contenidos. http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos (2010-04-01).

Xapian (2010). The Xapian Project. http://xapian.org (01-04-2010).

Zazo, Ángel F.; G. Figuerola, Carlos; Alonso Berrocal, José Luis; Rodríguez, Emilio (2005). Reformulation of Queries Using Similarity Thesauri // Information Processing & Management. 41:5 (2005) 1163-1173.

Zazo, Ángel F.; G. Figuerola, Carlos; Alonso Berrocal, José Luis; Rodríguez, Emilio (2004). El Sistema de Recuperación {K}arpanta: Estudio de Usuarios a Través del Archivo de Registro. // Scire.10:2 (2004) 63-76.

Recibido: 2010.04.22. Revisado: 2010-07-06. Aceptado: 2010-07-16.