Motores de Búsqueda Web
Tarea 1

José Alberto Benítez Andrades

71454586A

Motores de Búsqueda Web
Máster en Lenguajes y Sistemas Informáticos - Tecnologías del Lenguaje en la Web
UNED
07/12/2010

7 de diciembre de 2010

Tarea 1

Enunciado del ejercicio

Como introducción a los buscadores Web, os proponemos la lectura de este artículo: Arvind Arasu, Junghoo Cho, Hector García-Molina, Andreas Paepcke y Sriram Raghavan: Searching the Web. ACM Transactions on Internet Technology, vol. 1, num. 1, Agosto 2001, pp. 2-43. Una vez leido este articulo, debéis seleccionar en scholar.google.com tres artículos recientes (2005 o posterior) que lo citen y que tengan el mayor impacto posible, y a partir de ellos discutir tres factores en los que la investigación actual se sitúa mucho más allá de lo que se plantea en el artículo original.

- Después de realizar la lectura del artículo *Searching the web* he seleccionado los tres artículos siguientes, que lo citan en sus referencias:

1.*Using PageRank to Characterize Web Structure*

http://akpeters.metapress.com/content/6p488374j8h21088/

2.*Detecting Near-Duplicates for web crawling*

http://portal.acm.org/citation.cfm?id=1242592

3.*IRLbot: Scaling to 6 Billion Pages and Beyond*

http://portal.acm.org/citation.cfm?id=1541822.1541823

1. Resumen y conclusión del artículo *Searching the Web*

En el artículo *Searching the web*, se tratan diferentes puntos sobre el rastreo web, el almacenamiento de páginas, la indexación y el uso de diferentes técnicas para el diseño e implementación de una serie de componentes.

En primer lugar, realiza un pequeño resumen del diseño del motor de búsqueda en el año 2001. Posteriormente introduce una arquitectura de un motor genérico del cual se examinarán los distintos componentes. Trata también descriptores de categorías, servicios de

información, sistemas y software, y algoritmos de diseño e implementación. También habla sobre las palabras clave.

1.1. Introducción.

Los autores nos cuentan los distintos desafíos que se encuentran a la hora de realizar la creación de buenos motores de búsqueda, enumeran una serie de técnicas útiles. Concretamente, nos detallan las técnicas de IR (*Information Retrieval*), las cuales sirven para recuperar información de colecciones pequeñas, como por ejemplo, artículos de periódicos y catálogos de libros en bibliotecas.

Sin embargo, estas técnicas de IR, no son válidas para la web, ya que hay un gran volumen de ellas y usarían demasiados recursos, así que trata de mostrarnos técnicas nuevas como la *indexación* que produce escalabilidad a la hora de realizar rastreos web, el uso de técnicas de discriminación de webs con contenido irrelevante.También tiene en cuenta si la página es referida por otras páginas con unos términos concretos, lo que denotaría que esa página es importante con esos términos concretos.

En el año 2001 ya existían billones de páginas web en el mundo. En 1998, las páginas web pesaban entre 5 y 10 kbytes, 2 años más tarde duplicaron su tamaño.

En este artículo realizaron un estudio sobre medio millón de páginas, de las cuales, el 23% se actualizaban a diario, el 40% tenían dominios .com, el 28% funcionaban como núcleo y otro 44% podían enlazarse con el núcleo, pero no podían ser alcanzadas desde él.

Nos explican el funcionamiento de los *crawlers*, que son pequeñas aplicaciones encargadas de rastrear un repositorio de páginas web, observando los cambios que hay en ellas, incluyendo nuevas páginas que encuentran, etc. A través de complejos algoritmos de rastreo. Cuando el rastreador completa un ciclo, ya sabe qué páginas son las que debe rastrear y cuáles no. La optimización de recursos se consigue gracias a la indexación.

1.2. Rastreo de páginas web

En el segundo punto del artículo, nos resume las distintas funciones que debe hacer el rastreador a la hora de recoger la información de todas las webs. En primer lugar, las páginas pasan por un módulo de rastreo que recupera las páginas para un análisis posterior por el módulo de indexación. A través de un conjunto de webs que son recibidas, se asigna una prioridad a cada una de ellas para que sean analizadas.

Surgen problemas debido al gran tamaño y al constante cambio en las webs: 1) ¿ Qué páginas deben ser descargadas por el rastreador ? y 2) ¿ Cómo debe actualizar las webs el rastreador ?

El problema 1 se soluciona con prioridades por fracciones web, y el problema 2, que va unido al primero, se soluciona revisitando las webs que más cambien, porque si no se perderá mucho tiempo.

Otros problemas que surgen son por ejemplo: 3) ¿ Cómo reducirse la carga en las páginas visitadas? y 4) ¿Cómo debe el proceso de rastreo ser paralizado?. El primer problema, es bastante grave, ya que, el rastreo de webs consume mucho ancho de banda y mucha cpu, con lo cual, se deben minimizar los recursos al máximo. El problema 4) se soluciona mediante la paralelización de procesos.

Un tema que también es tratado de manera bastante amplia en el artículo es el método de selección de páginas, en el cual tratan distintos modelos de importancia, por popularidad, interés, ubicación , etcétera. Unido a esto, vienen los modelos de rastreo, concretamente trata sobre dos: rastreo y parada, rastreo y parada con umbral.

Además, la actualización de las páginas, o la "frescura" de los enlaces que hay en el buscador y estrategias de actualización son otros temas derivados en este punto.

1.3. Almacenamiento: Desafíos y repositorios.

En el tercer punto del artículo, trata sobre la escalabilidad que debe haber en los repositorios de almacenamiento de colecciones grandes páginas web.

Esto conlleva una serie de desafíos, ya que, el repositorio gestiona una gran colección de objetos de datos (en este caso páginas web), similar a sistemas de ficheros o a bases de datos. Surgen problemas como: escalabilidad, acceso dual, grandes actualizaciones a granel y páginas obsoletas.

1.4. Indexación: Estructuras de índices, desafíos, particionamientos, sistemas de indexación de textos.

Existe un módulo de análisis que crean una variedad de índices. El análisis produce enlaces, índices de texto e índices de utilidad. Las etiquetas html *H1, H2* o *<b>* facilitan el trabajo a los analizadores para saber qué información es la más importante en una web.

Con los índices lo que se crean son unos grandes grafos que poseen nodos y enlaces muy importantes, que facilitan la comprensión de textos por parte de los rastreadores

1.5. Ranking y análisis de enlaces: PageRank, algoritmos HITS y otras técnicas.

Al ser tan grande internet, la manera de evaluar qué web es más importante, es complejo. Así que se crearon distintos algoritmos que evalúan la importancia de una página web en función de diferentes parámetros.

Por un lado está el PageRank, que tiene variaciones: el PR Simple y el PR Práctico. El PageRank depende de la importancia de las webs, y los enlaces que existen desde ella y hacia ella, a más importante sea una web, más PageRank transmite a las webs que él mismo enlaza.

El otro algoritmos, HITS, se encarga de identificar, dada una consulta, un conjunto de páginas web naturales o páginas web de autoridades.

2.Breve resumen de los 3 artículos seleccionados.

2.1.Using PageRank to Characterize Web Structure

Los autores comentan en el *Abstract* sobre la existencia de una ley que sigue el PageRank analizando diferentes modelos que han creado a lo largo del experimento.

Comentan sobre la importancia de los algoritmos de escalabilidad web, comunidades de minería, etcétera, que dan lugar a la ley de PageRank que se rige por una serie de exponentes. Respecto a la distribución de los grados de PageRank existen dos puntos de vista: Mecanismos de ranking (ordenando las webs por popularidad) y el punto de vista de la teoría de grafos. El 1º más famoso es el PageRank.

La web es un grafo cuyos nodos son páginas html, poseen un grado entrante y un grado saliente. El grado entrante son los hiperlinks que apuntan hacia el nodo, y los salientes, los hiperlinks a los que se accede desde el nodo.

El PageRank asigna un valor real positivo a cada web, si una web *q* es referenciada desde muchos lugares, incrementa su PageRank. Existe un problema, muchos usuarios acceden a las webs de forma totalmente aleatoria, lo que dificulta que el PageRank sea correcto.

Existen 3 modelos de grafos en la web: Modelos de grado, modelos de PageRank y el modelo híbrido, que mezcla los 2 algoritmos. Después de experimentar con todos los modelos, han llegado a la conclusión de que el que mejor funciona es el de PageRank, ya que, los otros dos modelos son demasiado aleatorios.

2.2. Detecting Near-Duplicates for Web Crawling

Este artículo trata uno de los problemas más importantes existentes a la hora de rastrear las webs, ya que, incrementa mucho los recursos que debe utilizar el ordenador encargado de rastrear las webs, **la duplicidad de contenido**.

Existen rastreadores genéricos, y rastreadores de documentos concretos o de una temática concreta, es fácil encontrar documentos duplicados por plagio, pero para poder localizar estos y disminuir su tamaño, existen distintas técnicas, aquí trata sobre todo tres: el método Charikar, el algoritmo de mínima distancia de Hamming y los Road-Map.

El algoritmo de Charikar trata de transformar vectores grandes en huellas pequeñas. Una web es un conjunto de características con un peso determinado cada uno. Un conjunto de características es un vector grande, que con el método de *simhash* se reduce a una huella pequeña.

Con el método simhash existen dos problemas, la huella de un documento es un hash de sus características, con lo que, dos documentos similares, poseen una huella muy parecida.

Para tratar el problema de la distancia de Hamming, se comentan distintos tipos de soluciones y diseños para poder arreglarlo, mediante una serie de tablas que poseen datos de los documentos y permutaciones de los mismos. Además de esto, existen unos métodos de compresión de huellas que reducen en más de un 50% la información.

Hay distintos casos que provocan la duplicidad en la web: los distintos *mirrors* que existen y son necesarios para no sobrecargar un único servidor, el clustering de documentos relacionados, los directorios web, los plagios, la publicidad *spam* y las duplicidades de muchos dominios.

2.3.IRLBot: Scaling to 6 Billion Pages and Beyond

De los tres artículos que he seleccionado, me ha parecido el más instructivo, en él, se hacen una serie de experimentos con un servidor único, y distintas pruebas de rastreo con

múltiples técnicas y algoritmos distintos, mostrando los resultados que dan con cada uno y pudiendo ver qué técnicas son las mejores.

El IRLBot (así llamaron a su rastreador personal de webs) fue capaz de recoger 6.3 billones de páginas web en 41 días, mantuvo 7.6 billones de conexiones, descargó a una velocidad de 319mb/s (unas 1800 páginas por segundo), estuvo conectado con 117 millones de servidores y leyó más de 394 billones de enlaces con un grafo de 41 billones de nodos únicos.

Habla sobre la escalabilidad, la importancia de la rapidez de rastreo dependiendo del número de páginas, del almacenamiento que se utilice (discos duros RAID, caché, memoria ram...).

Trata como temas importantes, los posibles fallos en el rastreo debidos a los servidores DNS, que puedan fallar en un momento determinado, y el problema grave del SPAM, en el cual, muestran una serie de algoritmos y técnica para poder evitarlo, como el BEAST (Budget Enforcement Anti Spam Tactic). También nombra los famosos cuellos de botella que se forman a la hora de hacer rastreos.

Destaca también el DRUM (Disk Repository With Update Management) que es la técnica que mejor funciona para el rastreo de webs (mejor que el Mercator-B). Y también comenta el estado del PageRank después de finalizar su rastreo por todas las webs. Además, trata también el tema de la duplicidad de contenido en las páginas web.

3. Conclusiones finales en comparación con el primer artículo *Searching the Web.*

Después de buscar tres referencias más actuales sobre la búsqueda en la web y que además referencien este artículo, he podido observar los siguientes detalles:

- Velocidad, reducción de recursos y capacidad de almacenamiento: Tanto en el artículo que hemos tenido que leer, del año 2001, como en los 3 artículos más recientes que he seleccionado de scholar.google.com, he observado que dan una importancia suprema a la velocidad de rastreo, a la minimización de recursos a la hora de realizar estos rastreos (intentar consumir poco ancho de banda, poca CPU y poca memoria RAM) e intentar también tener el mayor número de información en el menor espacio posible. En los últimos artículos,

cuentan una serie de algoritmos de mejora de rastreo, que han conseguido optimizar al máximo la velocidad, los recursos y sobre todo el almacenamiento, reduciendo incluso en un 85% el tamaño real de las páginas web, mediante algoritmos de compresión.

- Problemas típicos de rastreo: En el primer artículo, el problema más grave a la hora de realizar los rastreos, era encontrar las webs que tenían un contenido importante, ya que el usuario en muchas ocasiones, entra en webs de manera bastante aleatoria. Los problemas que han surgido nuevos, son por ejemplo el SPAM y la duplicidad del contenido de las páginas web, que en el último artículo que leí, comenta las soluciones que existen para indicar a los rastreadores, como por ejemplo, las redirecciones 301.

- El PageRank o la importancia de las webs: Hace unos años, se inventó un algoritmo para calcular la importancia de un sitio web, el cual estaba basado en la importancia de las webs que te enlazaban, sobre todo. Esto a lo largo de los años ha cambiado, y el PageRank, no es actualmente el valor que más observa un rastreador para calcular su importancia en internet, ya que, se comenzaron a crear una serie de modas que fomentaron el spamming, provocando que los buscadores bloquearan este tipo de webs y las penalizaran en cuanto a posiciones en google se refiere. Ahora se tienen más en cuenta otros factores como la pureza del código (si cumple estándares css, w3c, xhtml... etcétera) además de que esté correctamente redireccionado con un buen fichero htaccess.