Web Crawler en eva.fing.edu.uy

Recuperación de Información y Recomendaciones en la Web(2014)

Grupo 4

Andrea Estefan CI 4303805-6 andyestefan@gmail.com

Gerardo Fanjul CI 4542811-6 gerardo.fanjul@gmail.com

Instituto de Computación - Facultad de Ingeniería
Universidad de la República Oriental del Uruguay
Docente: Libertad Tansini

Tabla de contenido

Introducción

Buscar información concreta dentro de un sitio grande, por ejemplo eva.fing.edu.uy a veces se vuelve una tarea engorrosa por la cantidad de links y recursos existentes. Además obliga al usuario a situar el tema de interés dentro de alguna categoría y comenzar a navegar por los links hasta encontrar o no algún resultado.

Esto nos llevó a plantearnos la necesidad de construir un software que escanee las páginas y devuelva la información encontrada.

Nuestro objetivo es obtener una lista de recursos existentes en la página del eva de facultad de ingeniería, de manera ágil y automática, brindando la posibilidad de filtrar según el tipo del material encontrado.

Web Crawler

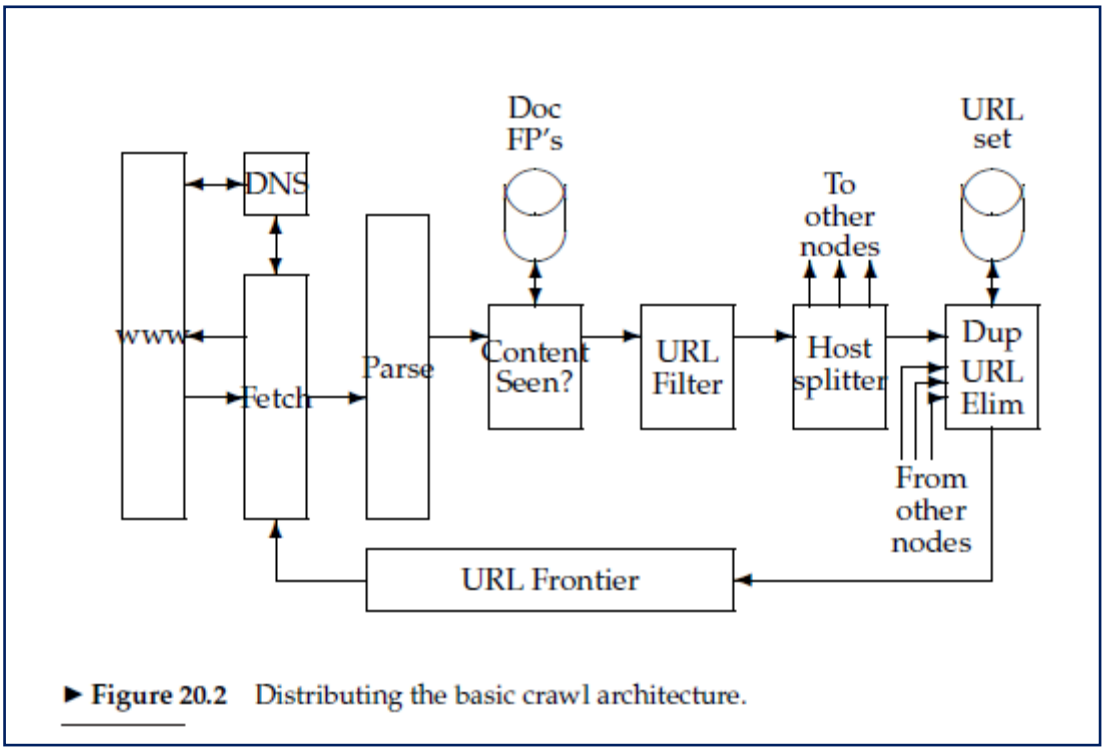
Definición

Web Crawler es proceso automatizado que explora Hipertexto.

El rastreo comienza con la definición de una o más URLs consideradas las semillas, se obtiene la página web con la que inicia el proceso y se analiza el hipertexto. Para esto se realiza un parseo, extrayendo texto y links existentes en la página. El texto extraído es indexado y los links encontrados son agregados a una lista de URLs llamada frontera. En la lista frontera se guardan aquellas URLs correspondientes a páginas que aún no han sido analizadas, al iniciar sólo se encuentran las restantes semillas y una vez analizadas se quitan del conjunto.

El proceso de rastreo web tiene diferentes usos, el más común es el asociado a los motores de búsqueda. Éstos usan rastreadores web para recopilar la información disponible en las páginas web públicas, indexar y reportar enlaces rotos.

Arquitectura



► **Figure 20.2** Distributing the basic crawl architecture.

Los elementos principales de la arquitectura son los siguientes:

1- URL Frontier: conjunto de URLs que aún no han sido descargadas y analizadas
2- Módulo de resolución de DNS
3- Fetch: módulo que utilice el protocolo http para recuperar la página correspondiente a una URL
4- Parser que extraiga la información necesaria de la página obtenida (texto y links)
5- Módulo de proceso de URLs que determine si un link ya fue visitado en una etapa anterior y resuelve su ingreso a la frontera aplicando si existen políticas específicas

Web Crawler en eva.fing.edu.uy

Implementación

Con el fin de agilizar la búsqueda de material en la página del eva de la facultad de ingeniería desarrollamos una aplicación que escanea las páginas y navega por sus links obteniendo los recursos buscados.

Nuestro algoritmo comienza con el análisis de la página semilla https://eva.fing.edu.uy/course/index.php?categoryid=7, página de InCo dentro del eva de la facultad de ingeniería.

El usuario define una palabra clave del tema que está buscando y selecciona el o los tipos de recursos que desea obtener. Como la página del eva posee muchos links, definimos una constante NIVEL, usada como tope máximo de la profundidad de navegación para que la recursión termine luego de una cantidad acotada de pasos.

El análisis comienza descargando la página HTML a partir de una URL, luego se obtienen todos los elementos href de la página.

Si el elemento obtenido posee la palabra buscada, se determina qué tipo de recurso es siguiendo las siguientes reglas:

Regla	**Recurso**
eva.fing.edu.uy/mod/resource	formato PDF o PPT
eva.fing.edu.uy/mod/url	Link internos del eva
eva.fing.edu.uy/mod/page	Páginasexternas
eva.fing.edu.uy/mod/folder	Carpetas

Si coincide con alguno de los tipos buscados, se inserta la URL y el tipo del recurso dentro de la tabla Record.

En el caso de ser un link que no coincide con la palabra clave buscada, y esté dentro de algún curso (la url contenga el substring eva.fing.edu.uy/course) se continúa la recursión explorando este nuevo link encontrado.

Buscadores automáticos

Aquellos que a partir de cierta información entregada en lenguaje natural o en alguna especificación puede deducir y recuperar la información que uno está buscando.

Objetivo: Encontrar los documentos que contengan la o las palabras claves introducidas. Habitualmente localiza las páginas Web que mejor se adapten a las palabras introducidas.

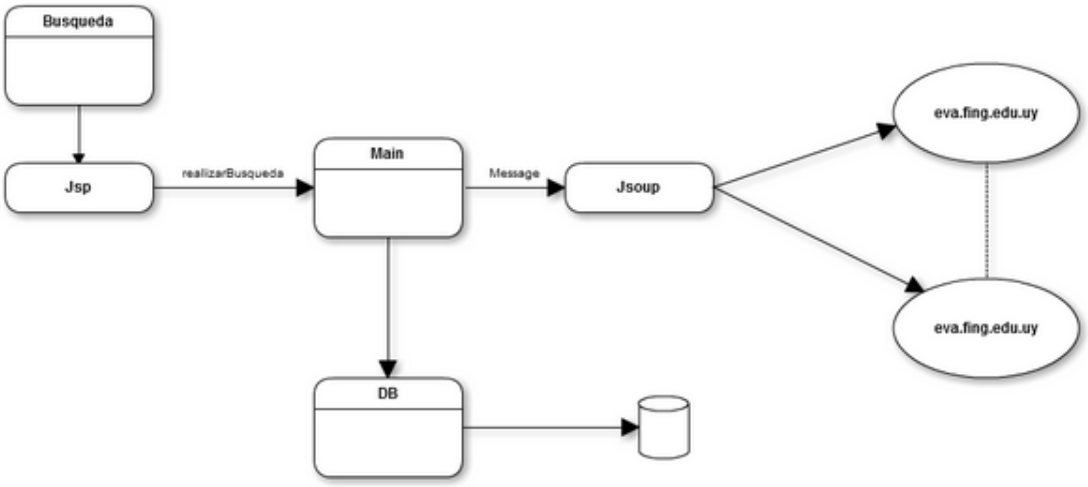
Si se quiere que la búsqueda localice también aquellos documentos donde las palabras tecleadas aparezcan como partes de otras palabras es necesario tener una base de datos con palabras "parecidas" o de cierta cercanía.

Para esto se creó una pequeña base de datos con palabras cercanas o parecidas, que al iniciar la búsqueda lo hacemos por todas las palabras que sean parecidas a la seleccionada.

Tolerancia a errores

A modo de ejemplo se implementa una pequeña extensión del vocabulario compuesta por sinónimos, correcciones ortográficas y/o conceptos relacionados.

Arquitectura y Diseño



Búsqueda: Este elemento es el que se encarga de la interfaz de usuario, es quien recaba los datos introducidos por el usuario a través de Servlets de java y JavaBeans el cual le comunica la petición de html a la clase principal que se encarga de realizar la búsqueda.

Main: Es la entidad encargada de realizar la búsqueda propiamente dicha, conteniendo y ejecutando el algoritmo de búsqueda de sinónimos. Luego realiza el crawler con cada una de las palabras sinónimo de la solicitada.

El rastreo de información lo realiza a través de la librería Jsoup en la página eva.fing.edu.uy partiendo de una semilla inicial. A medida que visita páginas o links se van guardando en la base de datos y marcadas como visitadas.
Se realiza un parseo en búsqueda del recurso requerido y en caso de encontrar un macheo, este se almacena en la base de datos para luego ser mostrado en la interfaz web.

DB: Este módulo se encarga de realizar la comunicación con la base de datos a través de jdbc driver. Es la encargada de crear los prepareStatements que se utilizarán en consultas con el fin de optimizar los accesos a la base de datos.

Esquema de la base de datos

CodTipoRecurso contiene los tipos de recursos, pueden ser pdf, URL, página o carpeta.

En Record se van almacenando tanto los links ya navegados y escaneados como los links que son recursos encontrados.

Sinónimos guarda una pequeña extensión del lenguaje a modo de ejemplo. Las palabras son categorizadas por grupos.

codtiporecurso

Columna	Tipo	Nulo	Predeterminado	
id	int(11)	No		
descripcion	text	No		

record

Columna	Tipo	Nulo	Predeterminado	
RecordID	int(11)	No		
URL	text	Sí	*NULL*	
RESOURCE	text	Sí	*NULL*	
idTipoRecurso	int(11)	Sí	*NULL*	

sinonimos

Columna	Tipo	Nulo	Predeterminado	
id	int(11)	No		
palabra	text	No		
grupo	int(11)	No		

Herramientas

Éste web crawler es implementado en Java, utilizando la librería JSoup 1.8.1.

Jsoup es una api open source desarrollada por Jonathan Hedley que proporciona herramientas para analizar HTML. Permite parsear, manipular y extraer datos de un HTML a partir de una URL, un archivo o un simple string.

La aplicación corre en un servidor Tomcat v 8.0, para la interfaz web se utilizó jsp con la ayuda del framework de diseño bootstrap.

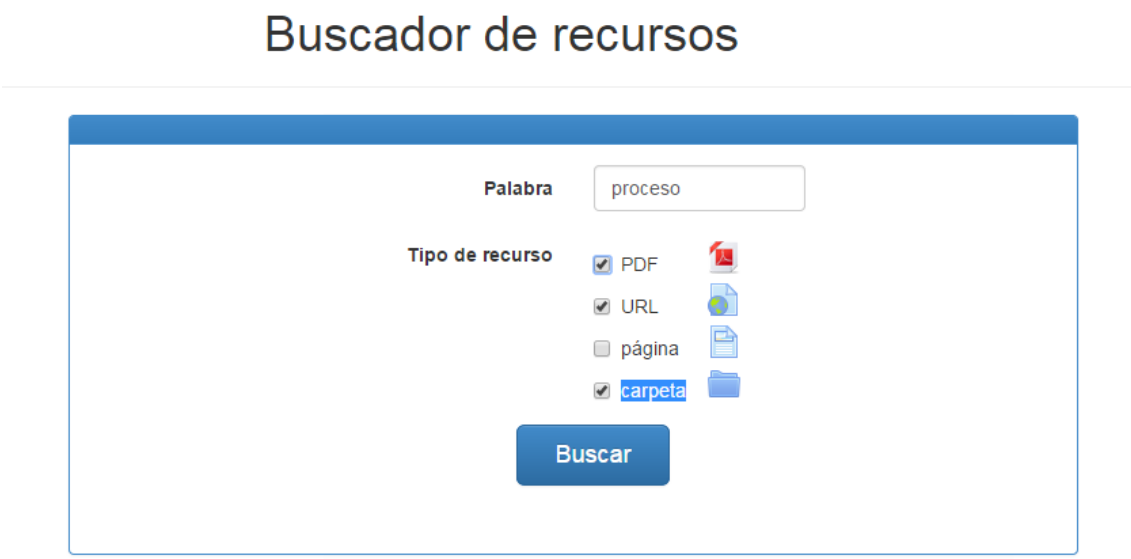
Para almacenar datos utilizamos MySql como gestor de nuestra base de datos relacional.

Como complemento destacamos la utilización de Maven integrado al IDE Eclipse Luna y Java 1.7

Caso de prueba

Se configura la página cursos dentro del eva como semilla del web crawler

A continuación se selecciona la palabra a buscar y tipos de recursos



Como la palabra **proceso** se encuentra dentro de nuestra pequeña base que extiende el vocabulario, no solo se van a mostrar los resultados para proceso, sino que también para todas las palabras pertenecientes a su grupo:

Dentro del grupo 1 se encuentran las palabras relacionadas con proceso

Grupo 1 = [Proceso, Procesos, proceso, procesos]

El web crawler comienza a escanear las páginas buscando la presencia de alguna palabra perteneciente al conjunto requerido.

Cada link visitado es almacenado dentro de la tabla record

RecordID	URL
NULL	https://eva.fing.edu.uy/course/index.php?categoryi...
NULL	https://eva.fing.edu.uy/course/index.php?categoryi...
NULL	https://eva.fing.edu.uy/course/index.php?categoryi...
NULL	https://eva.fing.edu.uy/course/index.php?categoryi...
NULL	https://eva.fing.edu.uy/course/index.php?categoryi...
NULL	https://eva.fing.edu.uy/course/view.php?id=212
NULL	https://eva.fing.edu.uy/course/view.php?id=5
NULL	https://eva.fing.edu.uy/course/view.php?id=767
NULL	https://eva.fing.edu.uy/course/view.php?id=767&mym...
NULL	https://eva.fing.edu.uy/course/view.php?id=767&mym...
NULL	https://eva.fing.edu.uy/course/info.php?id=767
NULL	https://eva.fing.edu.uy/course/info.php?id=767&mym...
NULL	https://eva.fing.edu.uy/course/info.php?id=767&mym...
NULL	https://eva.fing.edu.uy/course/view.php?id=739
NULL	https://eva.fing.edu.uy/course/view.php?id=739&mym...
NULL	https://eva.fing.edu.uy/course/view.php?id=739&mym...
NULL	https://eva.fing.edu.uy/course/info.php?id=739
NULL	https://eva.fing.edu.uy/course/info.php?id=739&mym...
NULL	https://eva.fing.edu.uy/course/info.php?id=739&mym...
NULL	https://eva.fing.edu.uy/course/view.php?id=725
NULL	https://eva.fing.edu.uy/course/view.php?id=725&mym...
NULL	https://eva.fing.edu.uy/course/view.php?id=725&mym...
NULL	https://eva.fing.edu.uy/course/info.php?id=725
NULL	https://eva.fing.edu.uy/course/info.php?id=725&mym...

Cada recurso encontrado es almacenado con su tipo:

RecordID	URL	RESOURCE	idTipoRecurso
NULL	NULL	https://eva.fing.edu.uy/mod/page/view.php?id=43863	3
NULL	NULL	https://eva.fing.edu.uy/mod/page/view.php?id=37247	3
NULL	NULL	https://eva.fing.edu.uy/mod/page/view.php?id=8586	3

Por último se presentan al usuario los datos obtenidos donde cada link lleva directamente al recurso encontrado

#	Recurso	Tipo
1	https://eva.fing.edu.uy/mod/resource/view.php?id=41099	
2	https://eva.fing.edu.uy/mod/resource/view.php?id=37402	
3	https://eva.fing.edu.uy/mod/resource/view.php?id=43812	

Configuraciones

El script de la base de datos se encuentra en el archivo webir.sql , el usuario y la contraseña para la conexión es [usuario = webir, pass= webir]

La credencial de página semilla se debe alojar en el directorio:

`C:\\-.fing.edu.uy.jks`

Por defecto la semilla del crawler es la página https://eva.fing.edu.uy/course/index.php?categoryid=7correspondiente al instituto de computación dentro del eva de la facultad de ingeniería.

Conclusiones y trabajo a futuro

Web crawler es una poderosa herramienta de búsqueda que puede agilizar la obtención de datos dentro de páginas con mucha navegación.

En nuestro pequeño ejemplo aplicado logramos obtener reglas básicas que nos permitieron identificar dentro de un HTML del eva un conjunto de recursos. Con estas reglas obtuvimos en cuestión de segundos material almacenado en el eva referente a un tema particular, sin la necesidad de conocer el contexto específico del mismo y ahorrándonos tiempo de navegación.

Como trabajo a futuro nos queda pendiente poder seleccionar una o varias semillas, para esto es necesario investigar el uso dinámico de credenciales para la conexión a páginas https.

Implementar un módulo de alta, baja y modificación de reglas de búsqueda dependiendo del tipo de recurso requerido.

En esta implementación se agregó una extensión del leguaje a modo de ejemplo, queda pendiente la investigación del uso de librerías que proporcionen sugerencias de palabras en español.

Referencias

http://jsoup.org/apidocs/

http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

http://getbootstrap.com/

https://eva.fing.edu.uy/course/index.php?categoryid=7

http://www.monografias.com/trabajos/buscadores/buscadores.shtml#ixzz3L1m0yVAf