Extracción de la Información de la Web

Máster Online en Ciencia de Datos

UCO ONLINE

Dr. José Raúl Romero

Profesor Titular de la Universidad de Córdoba y Doctor en Ingeniería Informática por la Universidad de Málaga. Sus líneas actuales de trabajo se centran en la democratización de la ciencia de datos (*Automated ML* y *Explainable Artificial Intelligence*), aprendizaje automático evolutivo y analítica de software (aplicación de aprendizaje y optimización a la mejora del proceso de desarrollo de software).

Miembro del Consejo de Administración de la *European Association for Data Science*, e investigador senior del Instituto de Investigación Andaluz de *Data Science and Computational Intelligence*.

Director del **Máster Online en Ciencia de Datos** de la Universidad de Córdoba.

Introducción a la Extracción de Hiperenlaces Web: *Web Crawling*

Introducción a *Web Crawling*

UCO ONLINE

Definición

- Los **rastreadores web** (*web crawlers*) – también, arañas (*spiders*) o robots – son programas que automatizan la descarga de páginas web
- Visitan multitud de sitios para recopilar información a analizar y minar, bien ***online*** (tras descarga) u ***off-line*** (tras descarga y almacenamiento)
- Las páginas web cambian frecuentemente, por lo que se requiere una actualización constante del rastreo
- Pueden enfocar la búsqueda de distinta forma:
  - **Rastreadores universales**: Buscan todas las páginas, indistintamente del contenido
  - **Rastreadores preferenciales**: Descargar páginas de un cierto tipo o temática

Aplicaciones de rastreadores

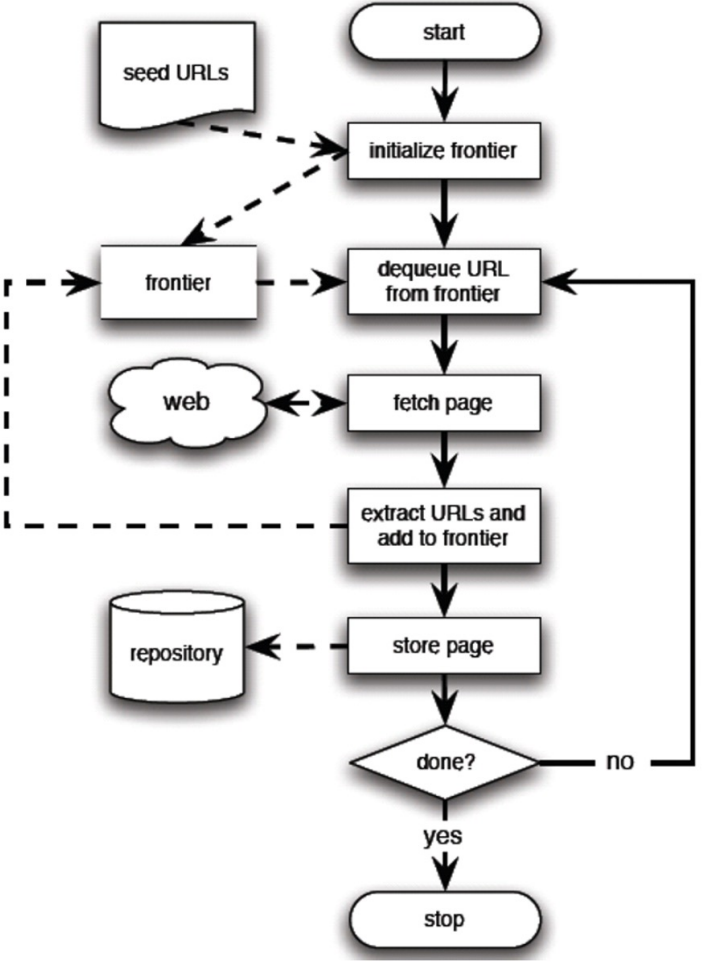
- En ***business intelligence*** las organizaciones necesitan recopilar información constante de sus competidores o potenciales colaboradores
- **Monitoreo de sitios Web** y páginas de interés, p.ej. para informar a una comunidad de que cierta información nueva ha aparecido
- Soporte a los **motores de búsqueda**:
  - Es su uso más extendido
  - Los rastreadores, junto con el *streaming*, consume gran parte del ancho de banda de Internet
- **Usos maliciosos**, como recopilación de emails para spamming o información personal para phishing

Algoritmo básico de rastreo

- Se inicia el proceso con páginas **semilla** (*seeds*), y sus links se utilizan para buscar (*fetch*) otras páginas
  - Las páginas enlazadas se visitan y se continúa con la extracción
  - Se repite el proceso hasta que se visiten un número adecuado de páginas, no queden páginas a visitar o se haya alcanzado un objetivo (**criterio de parada**)
- El algoritmo de rastreo es un algoritmo de búsqueda en grafos: páginas son nodos e hiperenlaces, las aristas
  - El modelo más básico sería un algoritmo secuencial → **ineficiente** (una búsqueda simultánea cada vez)

Algoritmo básico de rastreo

- La lista de URLs no visitadas se denomina **frontera** (*frontier*) → principal estructura de datos del rastreador
  - Se inicializa con las semillas
  - En cada visita, se extraen nuevas URL que se añaden a la lista.
  - La página se guarda en disco o bien se procesa extrayendo información y términos, que se guardan en memoria
  - Rara vez la frontera se vacía → es necesario un **criterio de parada**
  - El mecanismo por el que se establece el orden de extracción de URLs de la frontera determina el tipo de rastreo que implementa el algoritmo



Algoritmo básico de rastreo

- **Rastreadores en amplitud** (*breadth-first crawlers*)
  - La frontera es una cola FIFO (toma URLs de cabecera, introduce en cola)
  - No implica aleatoriedad, ya que las páginas más populares es más probable que reciban más enlaces, por lo que serían introducidas en FIFO prontamente
  - Altamente correlado con *PageRank*
  - Se ve muy afectado por la elección de la semilla
  - El registro histórico de rastreo (***crawl history***) contiene el listado de URLs visitadas y la fecha de visita
    - El registro se puede guardar en disco, pero es frecuente que se mantenga en memoria para las comprobaciones de si se ha visitado o no una página ya
    - Es frecuente el uso de tablas Hash para su inserción y consulta ($O(1)$)
  - Requieren mecanismos para evitar URLs duplicadas en la frontera

Algoritmo básico de rastreo

- **Rastreadores preferenciales** (*preferential crawlers*)
  - La frontera es una cola de prioridad, para lo que se debe medir la prioridad del enlace no visitado
    - Si las páginas se visitan en el orden estricto especificado por el valor de prioridad en la frontera, se denomina **best-first crawler**
    - La cola de prioridad puede ser dinámica y reordenarse según la puntuación de URLs encontradas en cada nueva visita
  - La prioridad se fundamenta en propiedades **topológicas** (p.ej. grado de conectividad de página destino), de **contenido** (p.ej. similitud) o combinación de otras **propiedades medibles**
  - La elección de la **semilla es crítica** en este caso
  - Insertar una URL en la frontera requiere una **complejidad** de $O(log(F))$, siendo F el tamaño de la cola.
  - Eliminar una URL de la frontera requiere **complejidad** $O(log(F))$ y de la tabla Hash, $O(1)$

Introducción a la Extracción de Hiperenlaces Web: *Web Crawling*

Aspectos de implementación

UCO ONLINE

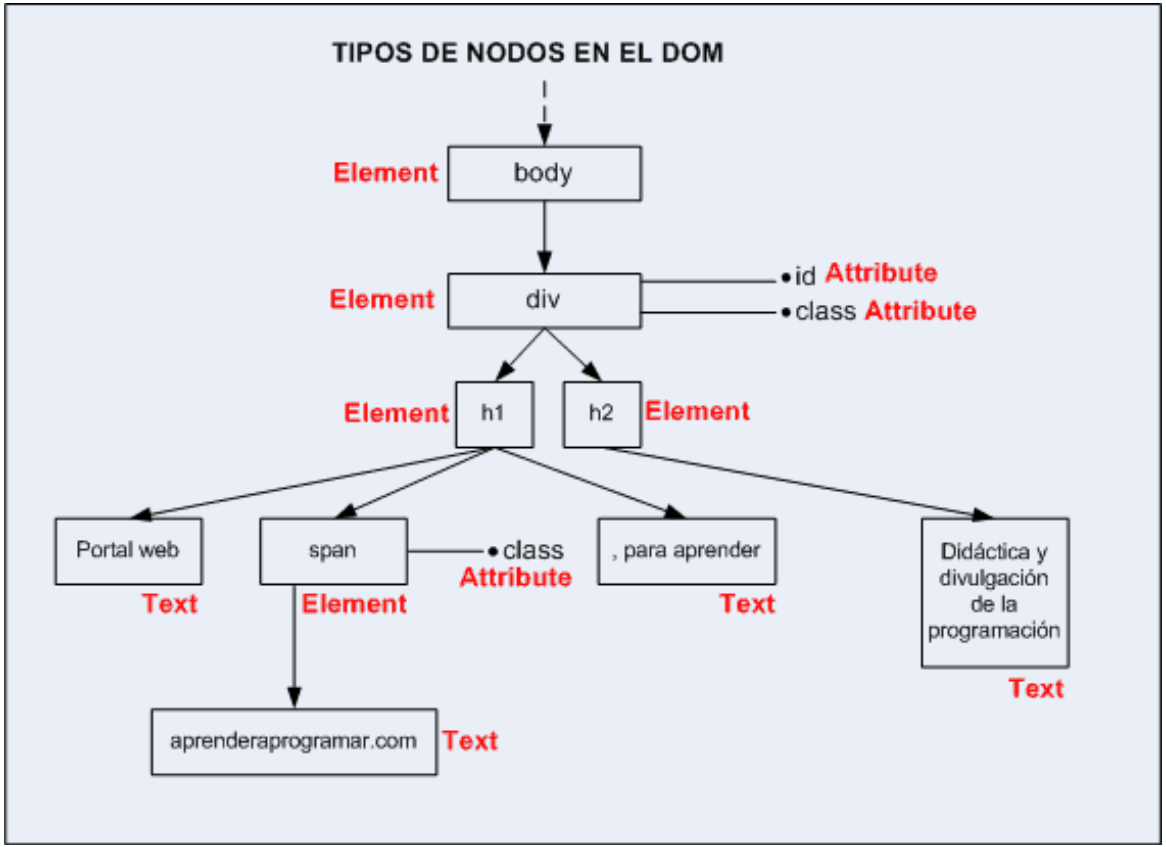
Fases de la implementación: Búsqueda

- El rastreador actúa como un cliente web:
  - Envía una solicitud HTTP (***HTTP request***) a un servidor y espera la respuesta (***HTTP response***)
  - Se debe considerar un tiempo de *timeout* para evitar bloquear el proceso
  - Se debe evitar la lectura de grandes páginas (según el caso), por lo que se suele limitar el tamaño de la descarga
  - Se parsean los códigos de estado HTTP de la cabecera de las respuestas
  - Se deben detectar los bucles de redirección (tabla Hash)
  - Puede ser necesario revisitar una página si ha pasado tiempo desde la última actualización de los datos

Fases de la implementación: Parseo

- Tras la descarga, la página se parsea (***HTTP payload***): identificación de *tags* y pares atributo-valor asociados dentro de la página
- Se extrae información según objetivos del rastreador: contenido (para indexación requerida por navegadores) y enlaces (para continuar el proceso de rastreo)
- Para analizar la página se realiza una búsqueda en profundidad del árbol DOM del documento HTML (**NOTA**: también pueden rastrearse otros tipos de documentos, JSON, CSV, etc.)
  - El uso de editores, programas no específicos (PowerPoint, Word, etc.) o el amplio conjunto de la población capaz de realizar un documento HTML complica considerablemente el trabajo de los rastreadores: etiquetas que faltan, instrucciones deprecadas, árbol mal construido, etc.
  - Lo más sencillo es la implementación de un rastreador de enlaces y/o texto de una página
  - Existen tecnologías que imposibilitan el rastreo (y posicionamiento): Flash, gráficos vectoriales, uso intensivo de Javascript y asincronía, etc.

Fases de la implementación: Parseo



Fuente: César Krall, aprenderaprogramar.com: CU01124E

Fases de la implementación: Limpieza

- Se aplican técnicas habituales de RI para limpieza de páginas visitadas
- Si se requiere extraer información del contenido, al parsear es útil eliminar las ***stopwords***, esto es, términos que no ayudan a la discriminación de las páginas por este contenido (conjunciones, artículos, etc.)
- Otra técnica que se aplica es el ***stemming*** (búsqueda de la raíz):
  - Las variantes morfológicas de un mismo término se concentrar en un único elemento raíz
  - Se requiere en el caso de los rastreadores preferenciales, en el que se considera las propiedades de contenido

Fases de la implementación: Extracción de enlaces

- Para la extracción de enlaces, se detectan las anclas (*anchor*) **<a>** y se toma el valor de la propiedad **href**
  - El texto del ancla puede ser muy descriptivo respecto al contenido de la página enlazada
  - Se requiere detectar el tipo de enlace para conocer si es interno/externo o el tipo de fichero al que se destina (algunos elementos pueden excluirse del rastreo) - p.ej. descartar el rastreo de enlaces a ficheros PDF
  - No siempre podemos confiar en el tipo de página por su extensión en el enlace (“.pdf”) → puede ser necesario enviar una solicitud **HTTP HEAD** e inspeccionar **content-type**

```
puede ver el <a href=“balance.html”>
balance de pagos de 2022</a>
```

Fases de la implementación: Extracción de enlaces

- También puede ser interesante filtrar según si la página es estática o dinámica
  - En el caso de las páginas dinámicas, el contenido (y los enlaces) se genera a partir de una base de datos
  - No siempre es sencillo detectar este tipo de páginas → a veces por la URL (p.ej. "/cgi-bin/", ".jsp"), otras por cabeceras que dejan los gestores de contenido, etc.
- Antes de añadir los enlaces a la frontera, convertir los enlaces relativos a absolutos tomando la URL base del servidor de la cabecera HTTP, de un meta-tag o del propio enlace siendo visitado

Fases de la implementación: Canonización

- Establecer la URL en una **forma canónica** (para frontera) establece un marco de comparación de enlaces en el rastreador (p.ej. para evitar que una página se visite varias veces)
- Requiere que todas las URLs sean absolutas
- Diferentes rastreadores pueden tener **distintas reglas** para la definición de formas canónicas
  - **Ejemplo**: a veces puede aparecer el número de puerto (incluyendo el 80) y otras no
  - No influyen estas diferencias siempre que las reglas sean consistentes para un mismo rastreador
- Puede ser necesario la aplicación de reglas heurísticas

Description and transformation	Example and canonical form
Default port number Remove	http://cs.indiana.edu:80/ http://cs.indiana.edu/
Root directory Add trailing slash	http://cs.indiana.edu http://cs.indiana.edu/
Guessed directory* Add trailing slash	http://cs.indiana.edu/People http://cs.indiana.edu/People/
Fragment Remove	http://cs.indiana.edu/faq.html#3 http://cs.indiana.edu/faq.html
Current or parent directory Resolve path	http://cs.indiana.edu/a/./../b/ http://cs.indiana.edu/b/
Default filename* Remove	http://cs.indiana.edu/index.html http://cs.indiana.edu/
Needlessly encoded characters Decode	http://cs.indiana.edu/%7Efil/ http://cs.indiana.edu/~fil/
Disallowed characters Encode	http://cs.indiana.edu/My File.htm http://cs.indiana.edu/My%20File.htm
Mixed/upper-case host names Lower-case	http://CS.INDIANA.EDU/People/ http://cs.indiana.edu/People/

Fases de la implementación: Repositorio de páginas

- La página debe indexarse/almacenarse para la aplicación final (p.ej. buscador web)
- Lo más sencillo: Tener un repositorio de páginas ya visitadas como ficheros individuales, donde cada página utiliza un valor hash para mapear su nombre con su URL (p.ej. MD5)
  - Puede ser inviable para rastreadores a gran escala por el consumo de recursos
- Más eficiente: Combinar muchas páginas en un único fichero
  - Requiere separar la tabla de búsqueda de la de mapeo con ficheros
- Más adecuado: Uso de base de datos indexada por **forma canónica** de URL
  - Por velocidad de acceso, mejor el uso de bases de datos incrustadas (p.ej. BerkeleyDB) al de sistemas relacionales

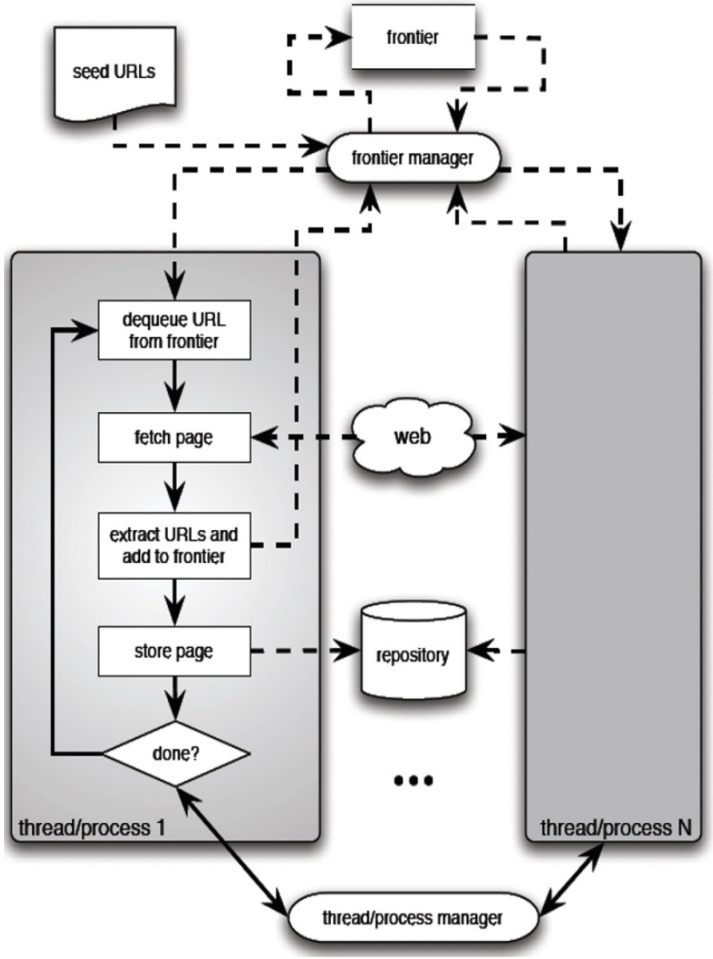
Introducción a la Extracción de Hiperenlaces Web: *Web Crawling*

Tipos de rastreadores

UCO ONLINE

Rastreadores universales

- Orientados a que buscadores de propósito general puedan mantener sus índices con un coste de rastreo asumible
- Indexan millones de consultas de usuario recibidas entre distintas actualizaciones de índice (actualizaciones incrementales)
- Difieren de los rastreadores en amplitud en su **rendimiento** (debe ser muy óptimo) y su **política** (cubren tantas páginas como puedan manteniendo tantos índices actualizados como sea posible – solución de compromiso)



Rastreadores focalizados

- Rastrean páginas solo de ciertas categorías en las que está interesado el usuario final del rastreador, y no de la Web entera → introduce un sesgo
- Ejemplo de aplicación: mantener categorizaciones o taxonomías Web (p.ej. directorio Yahoo! o *Open Directory Project - ODP*)
- Suelen hacer uso de un clasificador de texto:
  - El clasificador se construye a partir de un clasificador entrenado previamente (p.ej. con ODP)
  - El clasificador guía al rastreador seleccionando por preferencia aquellas páginas que más posiblemente pertenecen a la categoría de interés

    Pr(c|p) es la probabilidad de que una página rastreada p pertenezca a la categoría c
    Pr(top|p) = 1 por definición, siendo top la categoría raíz

  - En el estado del arte, la mayoría utiliza clasificadores bayesianos, si bien son mejorados por clasificadores basados en redes neuronales y SVM

Rastreadores temáticos

- Para tareas de rastreo preferencial no siempre hay disponibles suficientes páginas etiquetadas como para entrenar un rastreador focalizado
  - Se suele contar con un número limitado de semillas y la descripción de un tema de interés para un usuario o comunidad de usuarios
  - El **tema** lo pueden establecer una o más páginas de ejemplo, habitualmente las semillas, o una consulta de usuario de alcance limitado
  - En los rastreadores temáticos no se tiene un clasificador de texto
  - ☺ Pueden realizar el rastreo en tiempo real, mostrando al usuario los resultados por orden de puntuación (p.ej. similitud de contenido) o actualidad (p.ej. cabecera **`last-modified`**) → apropiado para aplicaciones que requieren resultados cambiantes y recientes
  - ☹ La búsqueda es lenta comparada con buscadores tradicionales
  - ☹ Los algoritmos de ranking en este caso no pueden aprovechar las medidas globales de prestigio (p.ej. PageRank) que sí disponen los buscadores tradicionales

Introducción a la Extracción de Hiperenlaces Web: *Web Crawling*

Conflictos con Web Crawling

UCO ONLINE

Trampas a rastreadores

- Muchos servidores Web, como Amazon, registran el comportamiento de sus clientes web para analizar el comportamiento de los compradores → crean una entrada en base de datos cada vez que hay un "*click*" o acceso
  - Un rastreador accediendo a enlaces de productos concretos puede distorsionar el análisis del servicio
  - El rastreador puede encontrar bucles o "infinitos" enlaces (URLs generadas dinámicamente apuntando a una entrada ya visitada, enlaces a comentarios de usuarios, etc.)
- Los rastreadores consumen tráfico, requieren generar datos inútiles para el servidor (páginas, caché, envío de cookies, entradas en BD) → considerados como un tipo de ataque DoS (***denial of service***)

Precaución al probar estos códigos: ¡¡Podemos ser bloqueados!!

Códigos de etiqueta

- Un primer conflicto está relacionado con la "**falta de educación**" que supone enviar decenas de solicitudes por segundo (o más) a un servidor que está tratando de responder a solicitudes de humanos interactuando → deterioro del servicio por sobrecarga
- Se debería **ser sincero en cuanto al tipo de agente** (rastreador) cuando se accede a servicios remotos → indicar el nombre y versión del rastreador en la cabecera HTTP **`User-Agent`**, así como el email del contacto en la cabecera **`From`**
- Seguir el **protocolo de exclusión de robots** (*Robot Exclusion Protocol*), que indica qué ficheros no deben ser accedidos por los rastreadores antes de buscar
  - Descargar el fichero opcional `/robots.txt`

```
User-Agent: *
Disallow: /
```

Códigos de etiqueta

- Seguir el **protocolo de exclusión de robots** es un aspecto ético voluntario:
  - Su incumplimiento puede tener ramificaciones legales
  - Algunos servidores pueden "banear" al cliente por su IP
  - Algunos rastreadores se disfrazan (User-Agent) de navegadores web e incluyen tiempos de interacción para dificultar la detección (no es ilegal pero sí poco ético)
  - **Cloaking**: Hay servidores poco éticos que detectan si el agente es un rastreador para devolver como respuesta páginas con distintos contenidos e hiperenlaces a las que mostrarían al usuario → objetivo: mejorar el ranking de su dominio en los buscadores
  - Los rastreadores deben tener algoritmos sofisticados para evitar enlazar publicidad de pago-por-click. Algunos servidores pueden "promover" que estos enlaces sean visitados evitando indicaciones semánticas al respecto
  - **Leyes de Copyright**: Algunos rastreadores poco éticos se dedican a realizar un "*mirroring*" del contenido de páginas de la competencia

¡Gracias!

UCO ONLINE