Informatics 225
Computer Science 221

Information Retrieval

Lecture 7

These course materials borrow, with permission, from those of Prof. Cristina Videira Lopes, Prof. Alberto- Krone-Martins, Addison Wesley 2008, Chris Manning, Pandu Nayak, Hinrich Schütze, Heike Adel, Sascha Rothe, Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. Powerpoint theme by Prof. André van der Hoek.

Ways of Acquiring Web Data

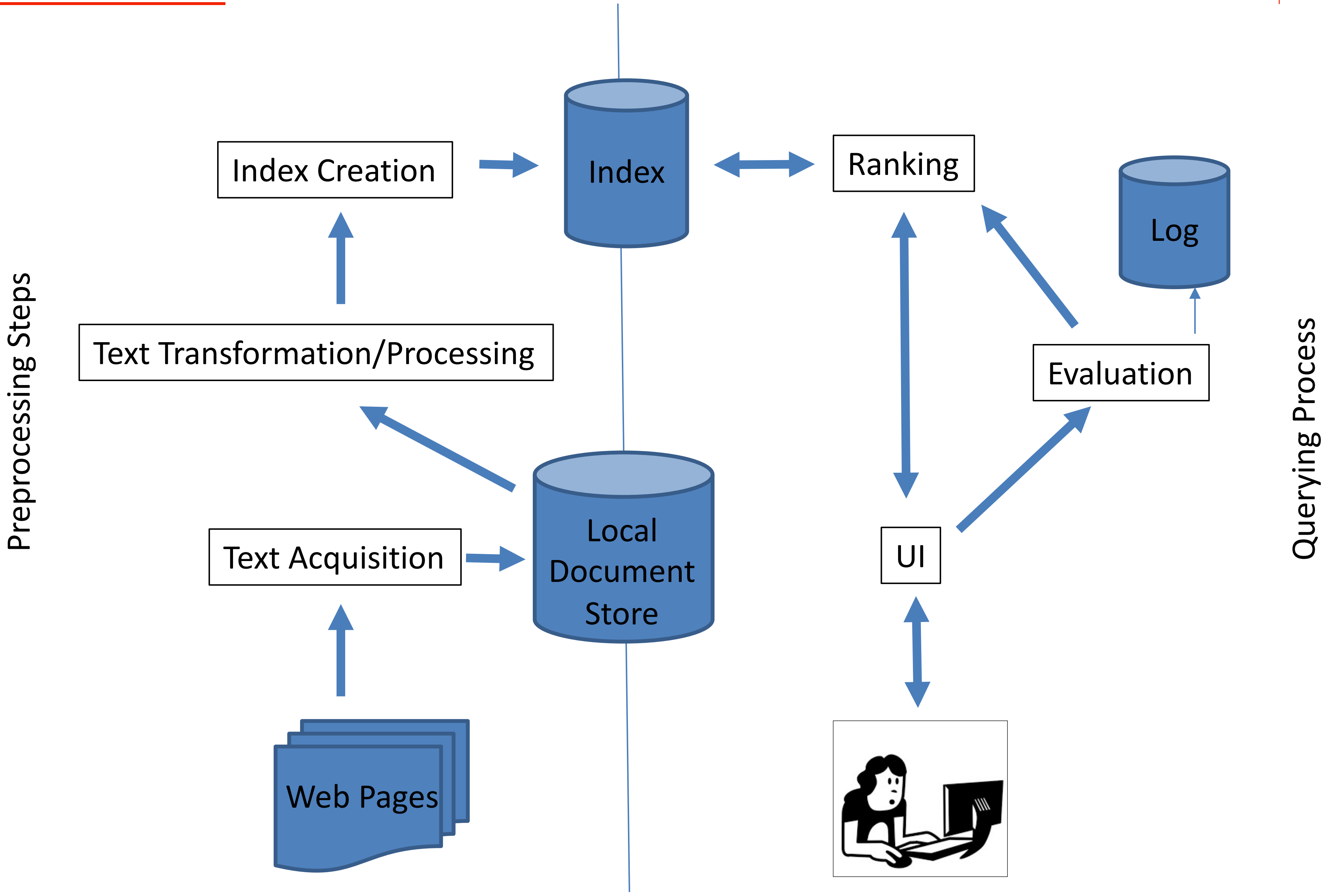
- Data dumps
- Web APIs
- Targeted downloads

- Web crawling ← last option

Crawling

Information Retrieval

Architecture



Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)

Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)
- Repeat until no more URLs in queue:

Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)
- Repeat until no more URLs in queue:
  - Get one URL from the queue

Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)

- Repeat until no more URLs in queue:
  - Get one URL from the queue
  - If the page can be crawled, fetch associated page

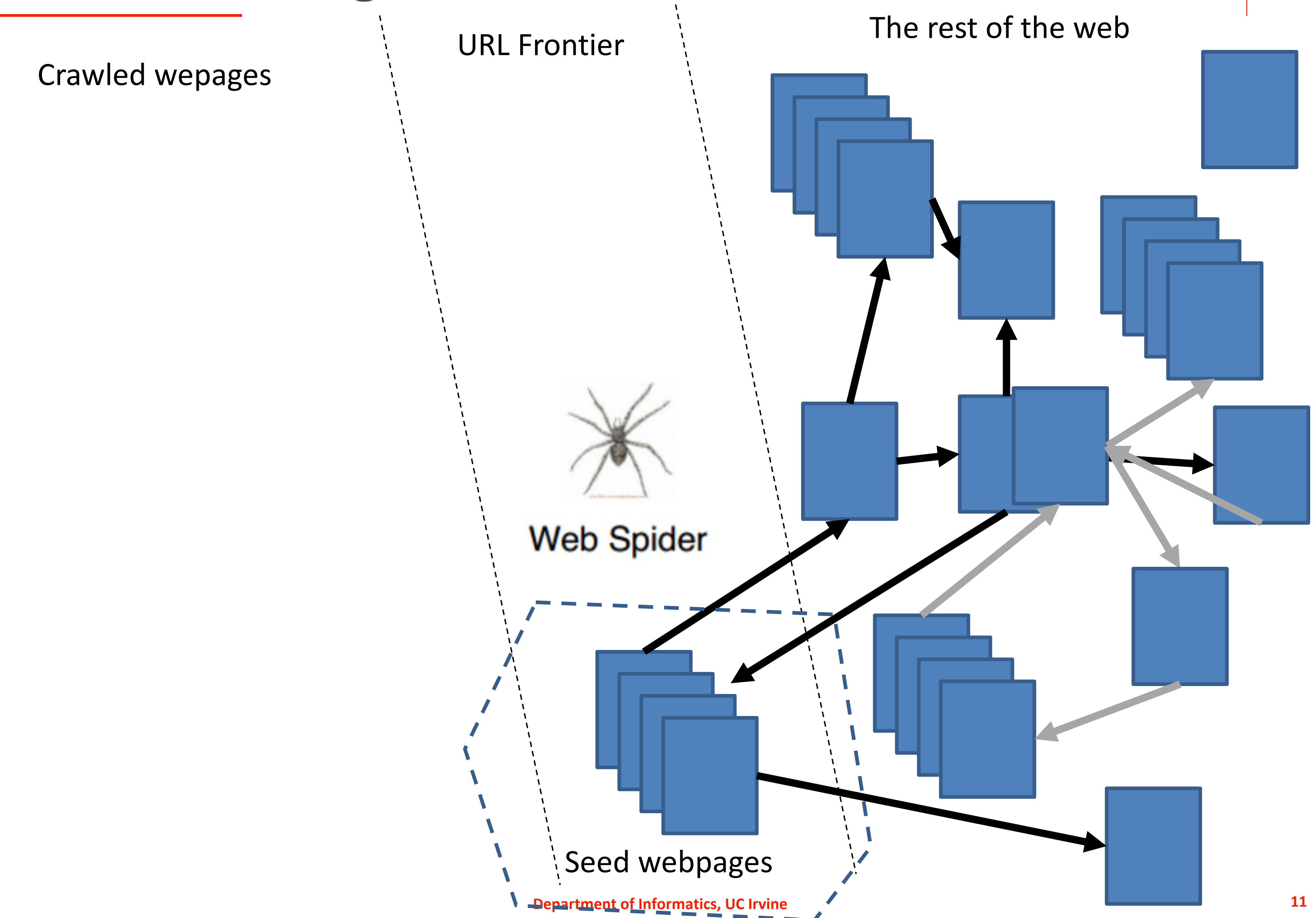
Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)
- Repeat until no more URLs in queue:
  - Get one URL from the queue
  - If the page can be crawled, fetch associated page
  - Store representation of page

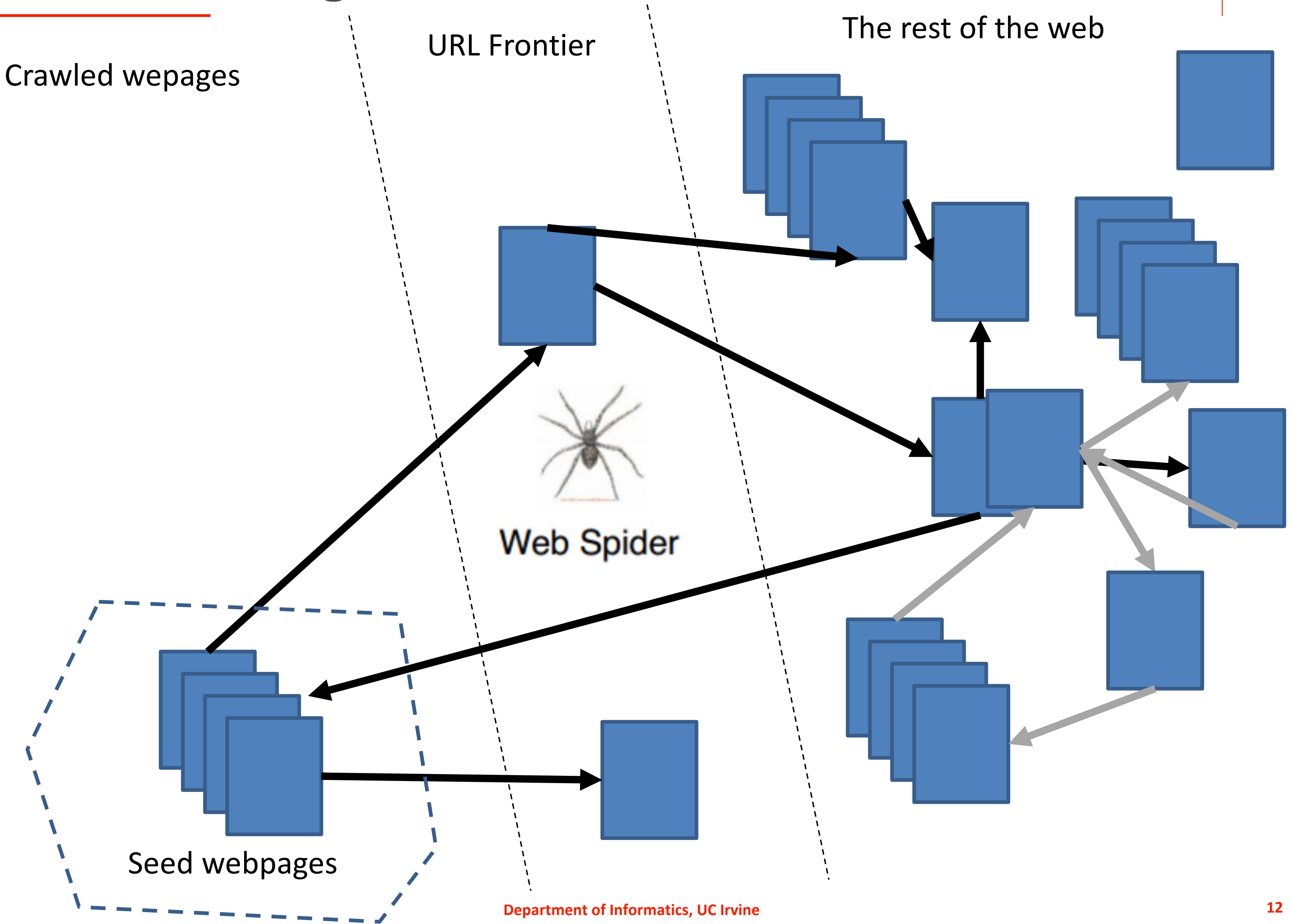
Basic Crawl Algorithm

- Initialize a queue of URLs (seeds)
- Repeat until no more URLs in queue:
  - Get one URL from the queue
  - If the page can be crawled, fetch associated page
  - Store representation of page
  - Extract URLs from page and add them to the queue
- Queue = “frontier”

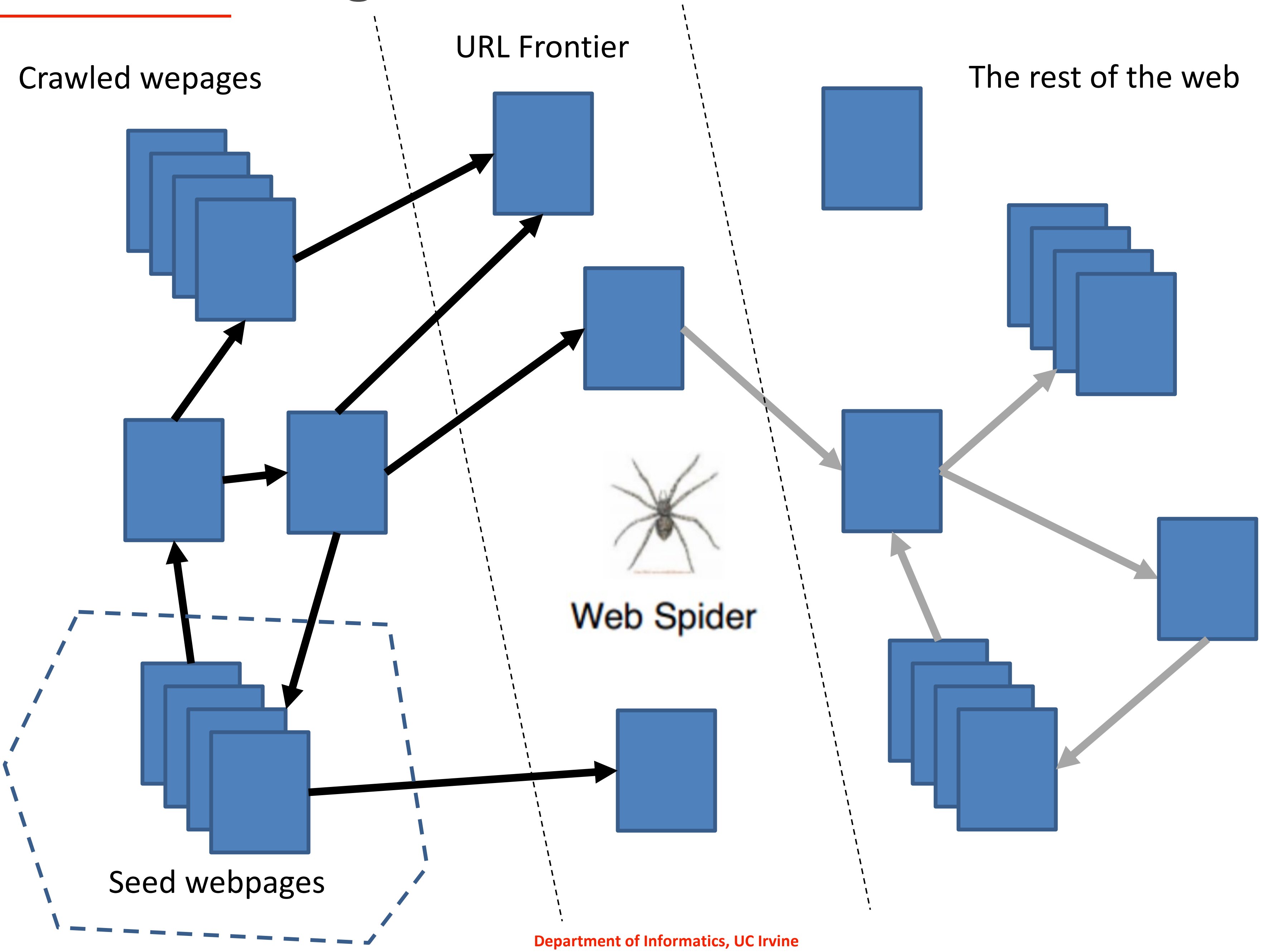
Basic Crawl Algorithm



Basic Crawl Algorithm



Basic Crawl Algorithm



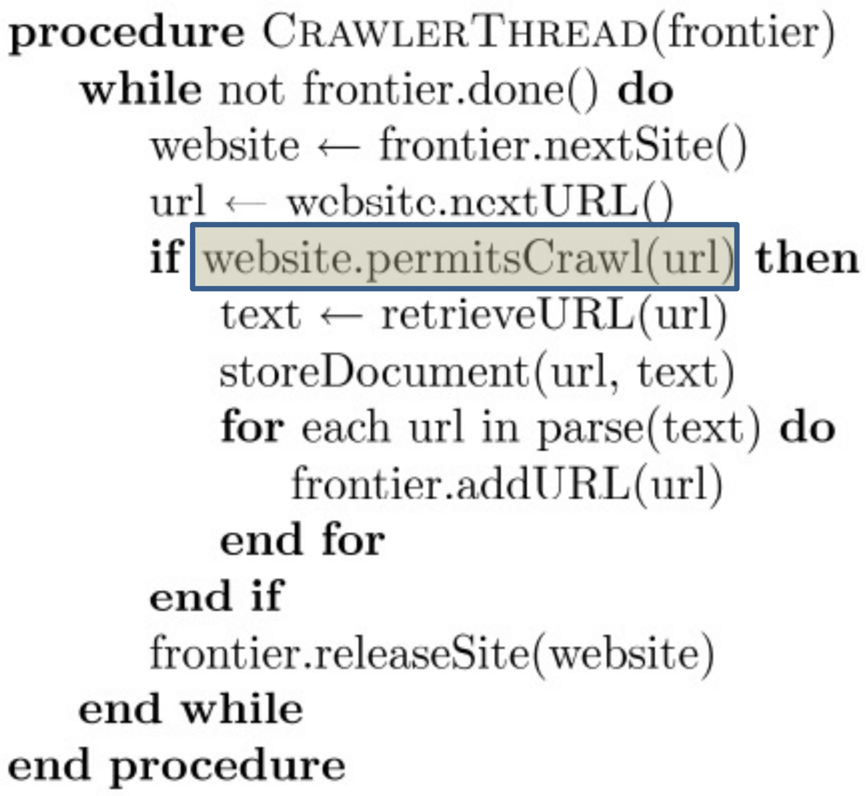
Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Pseudo Code



```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

But how do you know if the website allows you to crawl?

Permission to crawl

Permission to crawl

- Be polite: try to ask the website if you can crawl it first!
- Robots Exclusion Standard aka **robots.txt**
  - Sites may have that file at the root. Examples:
    - http://www.cnn.com/robots.txt
    - http://en.wikipedia.org/robots.txt
  - Very simple syntax (but no formal/official standard yet!):
    - http://www.robotstxt.org/robotstxt.html

Permission to crawl

Exclude all

```
User-agent: *
Disallow: /
```

Allow all

```
User-agent: *
Disallow:
```

Exclude all from a part of the site

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /junk/
```

Exclude a single robot

```
User-agent: BadBot
Disallow: /
```

Allow a single robot

```
User-agent: Google
Disallow:

User-agent: *
Disallow: /
```

http://www.robotstxt.org/robotstxt.html

Permission to crawl

Exclude all

```
User-agent: *
Disallow: /
```

Allow all

```
User-agent: *
Disallow:
```

There is a precedence rule:

**IF THERE IS A SPECIFIC PART FOR A NAMED BOT,
THE RULE THAT IS VALID FOR THAT BOT,
IS THE RULE THAT HAS THE NAME OF THE BOT!**

Exclude a single robot

```
User-agent: BadBot
Disallow: /
```

Allow a single robot

```
User-agent: Google
Disallow:

User-agent: *
Disallow: /
```

http://www.robotstxt.org/robotstxt.html

Permission to crawl

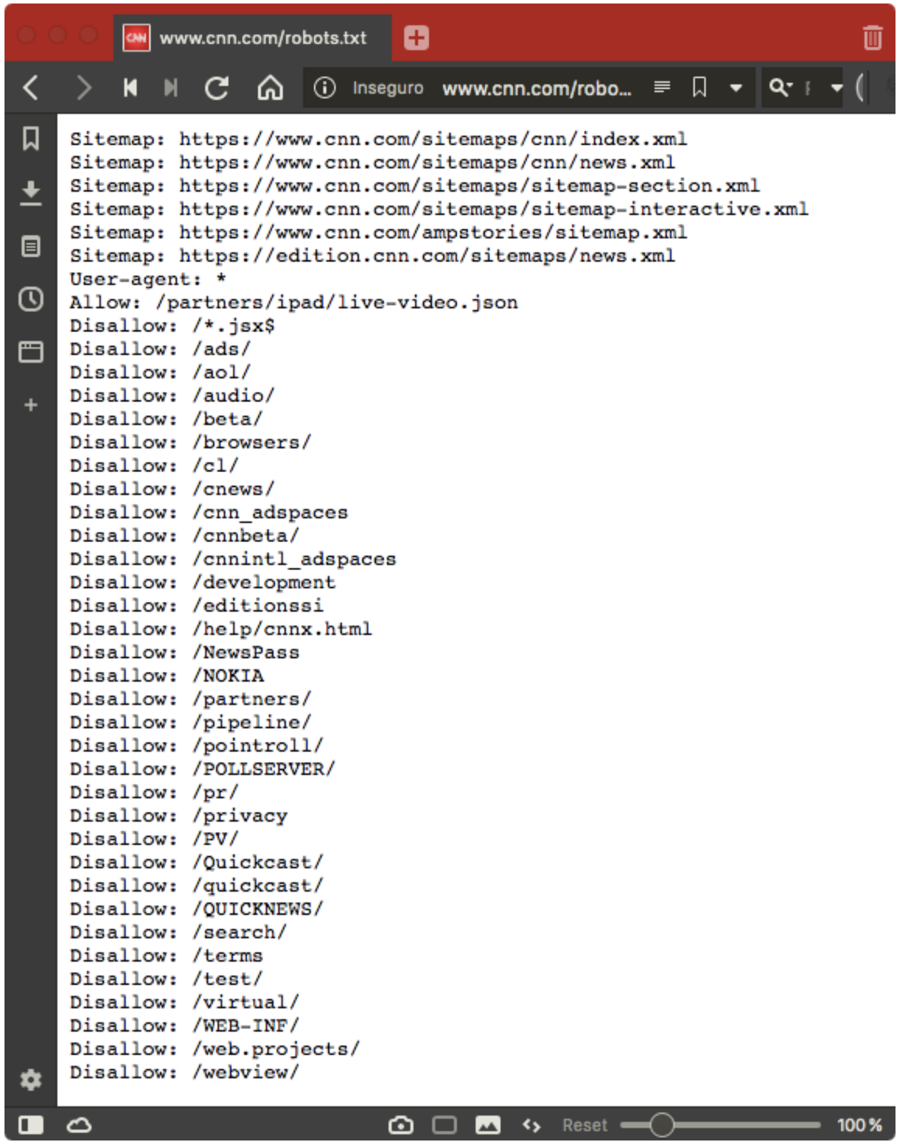
- Robots Exclusion Standard aka **robots.txt**
  - Sites may have that file at the root. Examples:
    - http://www.cnn.com/robots.txt
    - http://en.wikipedia.org/robots.txt
  - Very simple syntax (but no formal/official standard yet!):
    - http://www.robotstxt.org/robotstxt.html
  - **Honor basis!**
    - **It's not a security mechanism**

Information to crawlers

- Sitemaps (introduced by Google)
  - Also listed in robots.txt
  - Allow web masters to send info to crawlers. E.g.
    - Location of pages that might not be linked
    - Relative importance
    - Update frequency
- Example:
  - http://www.cnn.com/robots.txt

Information to crawlers

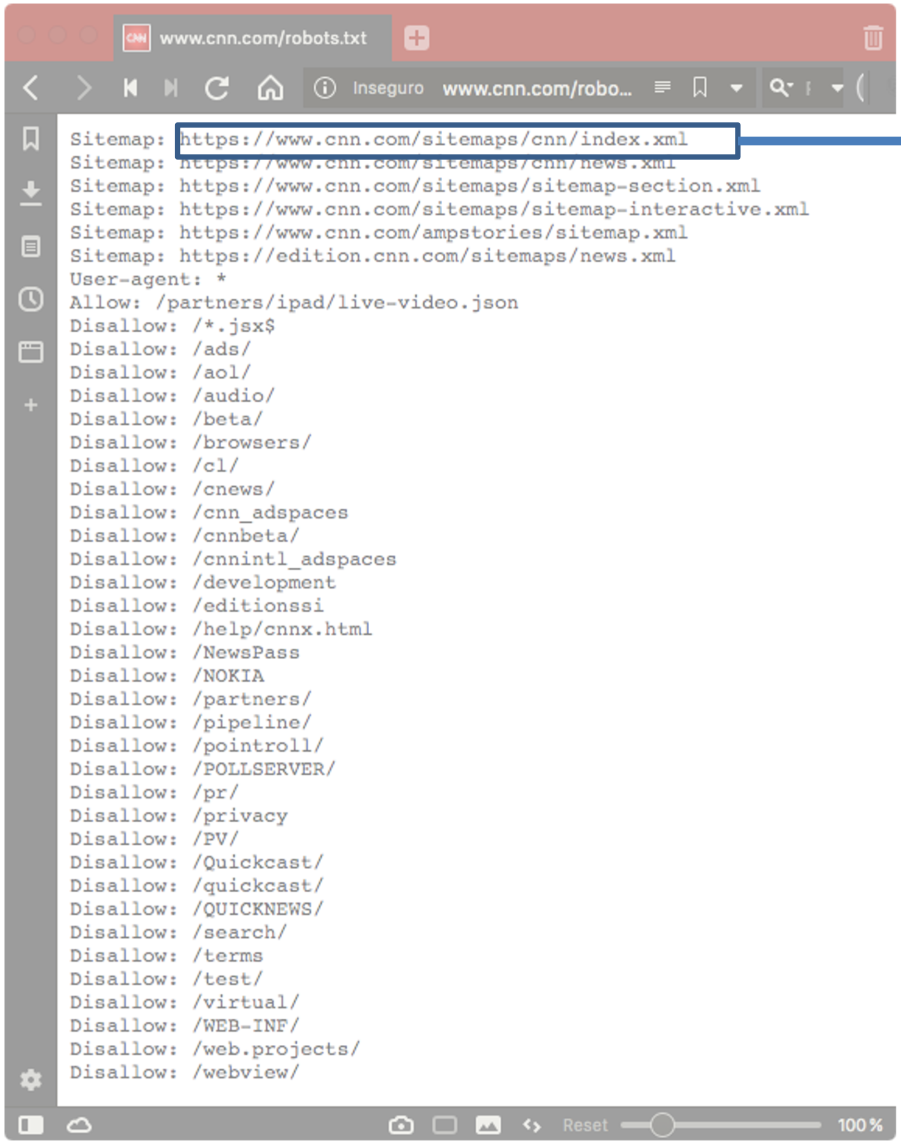
http://www.cnn.com/robots.txt



```
Sitemap: https://www.cnn.com/sitemaps/cnn/index.xml
Sitemap: https://www.cnn.com/sitemaps/cnn/news.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-section.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-interactive.xml
Sitemap: https://www.cnn.com/ampstories/sitemap.xml
Sitemap: https://edition.cnn.com/sitemaps/news.xml
User-agent: *
Allow: /partners/ipad/live-video.json
Disallow: /*.jsx$
Disallow: /ads/
Disallow: /aol/
Disallow: /audio/
Disallow: /beta/
Disallow: /browsers/
Disallow: /cl/
Disallow: /cnews/
Disallow: /cnn_adspaces
Disallow: /cnnbeta/
Disallow: /cnnintl_adspaces
Disallow: /development
Disallow: /editionssi
Disallow: /help/cnnx.html
Disallow: /NewsPass
Disallow: /NOKIA
Disallow: /partners/
Disallow: /pipeline/
Disallow: /pointroll/
Disallow: /POLLSERVER/
Disallow: /pr/
Disallow: /privacy
Disallow: /PV/
Disallow: /Quickcast/
Disallow: /quickcast/
Disallow: /QUICKNEWS/
Disallow: /search/
Disallow: /terms
Disallow: /test/
Disallow: /virtual/
Disallow: /WEB-INF/
Disallow: /web.projects/
Disallow: /webview/
```

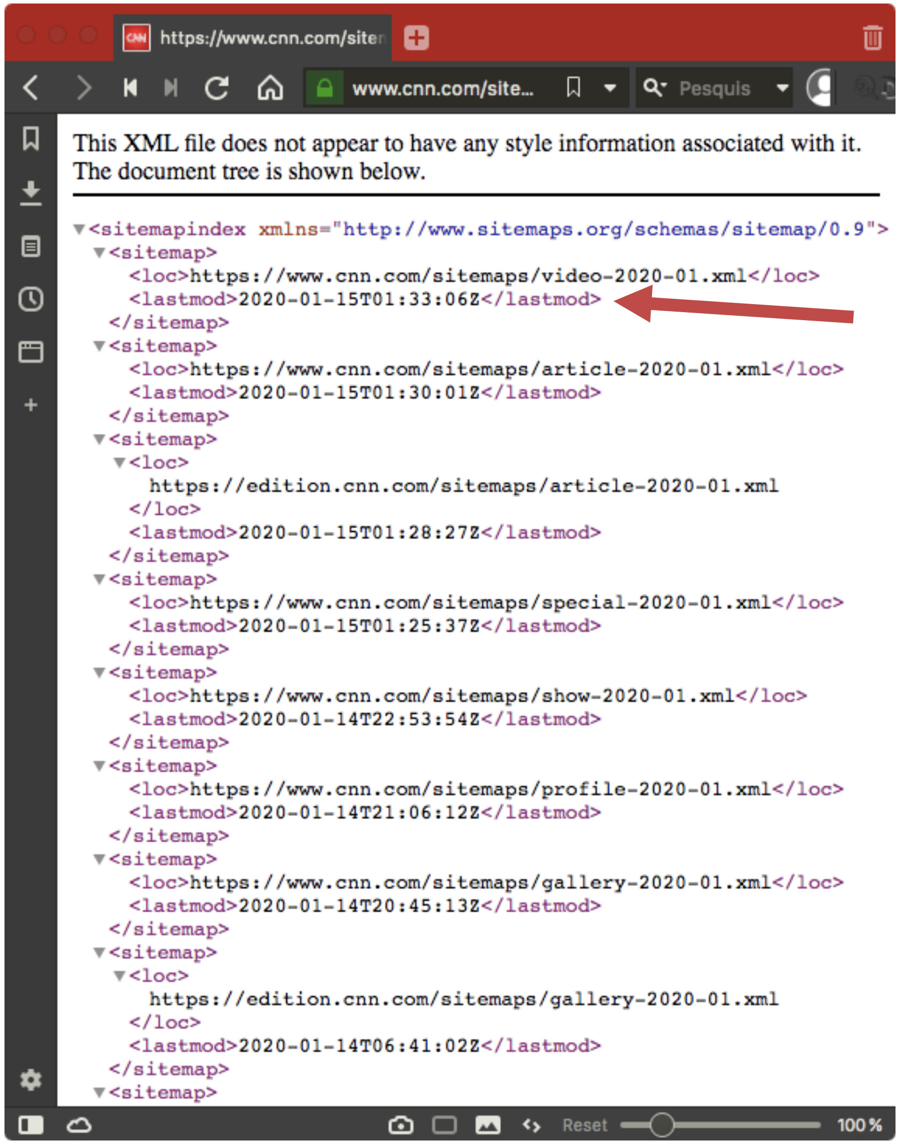
Information to crawlers

http://www.cnn.com/robots.txt



```
Sitemap: https://www.cnn.com/sitemaps/cnn/index.xml
Sitemap: https://www.cnn.com/sitemaps/cnn/news.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-section.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-interactive.xml
Sitemap: https://www.cnn.com/ampstories/sitemap.xml
Sitemap: https://edition.cnn.com/sitemaps/news.xml
User-agent: *
Allow: /partners/ipad/live-video.json
Disallow: /*.jsx$
Disallow: /ads/
Disallow: /aol/
Disallow: /audio/
Disallow: /beta/
Disallow: /browsers/
Disallow: /cl/
Disallow: /cnews/
Disallow: /cnn_adspaces
Disallow: /cnnbeta/
Disallow: /cnnintl_adspaces
Disallow: /development
Disallow: /editionssi
Disallow: /help/cnnx.html
Disallow: /NewsPass
Disallow: /NOKIA
Disallow: /partners/
Disallow: /pipeline/
Disallow: /pointroll/
Disallow: /POLLSERVER/
Disallow: /pr/
Disallow: /privacy
Disallow: /PV/
Disallow: /Quickcast/
Disallow: /quickcast/
Disallow: /QUICKNEWS/
Disallow: /search/
Disallow: /terms
Disallow: /test/
Disallow: /virtual/
Disallow: /WEB-INF/
Disallow: /web.projects/
Disallow: /webview/
```

https://www.cnn.com/sitemaps/cnn/index.xml



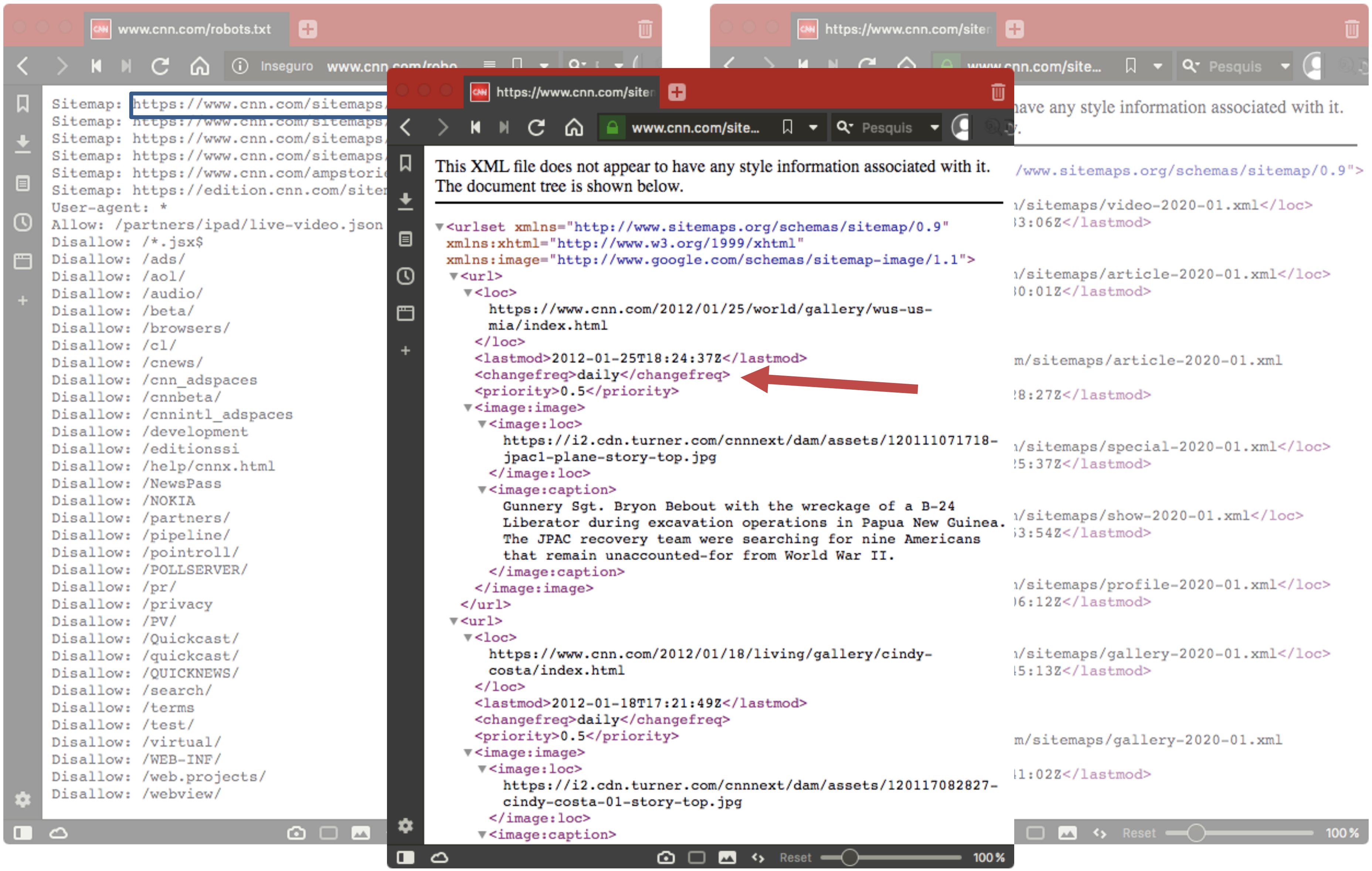
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/video-2020-01.xml</loc>
     <lastmod>2020-01-15T01:33:06Z</lastmod>
   </sitemap>
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/article-2020-01.xml</loc>
     <lastmod>2020-01-15T01:30:01Z</lastmod>
   </sitemap>
  ▼<sitemap>
    ▼<loc>
       https://edition.cnn.com/sitemaps/article-2020-01.xml
     </loc>
     <lastmod>2020-01-15T01:28:27Z</lastmod>
   </sitemap>
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/special-2020-01.xml</loc>
     <lastmod>2020-01-15T01:25:37Z</lastmod>
   </sitemap>
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/show-2020-01.xml</loc>
     <lastmod>2020-01-14T22:53:54Z</lastmod>
   </sitemap>
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/profile-2020-01.xml</loc>
     <lastmod>2020-01-14T21:06:12Z</lastmod>
   </sitemap>
  ▼<sitemap>
     <loc>https://www.cnn.com/sitemaps/gallery-2020-01.xml</loc>
     <lastmod>2020-01-14T20:45:13Z</lastmod>
   </sitemap>
  ▼<sitemap>
    ▼<loc>
       https://edition.cnn.com/sitemaps/gallery-2020-01.xml
     </loc>
     <lastmod>2020-01-14T06:41:02Z</lastmod>
   </sitemap>
  ▼<sitemap>
```

Information to crawlers

http://www.cnn.com/robots.txt

https://www.cnn.com/sitemaps/cnn/index.xml



```
Sitemap: https://www.cnn.com/sitemaps
Sitemap: https://www.cnn.com/sitemaps
Sitemap: https://www.cnn.com/sitemaps
Sitemap: https://www.cnn.com/sitemaps
Sitemap: https://www.cnn.com/ampstori
Sitemap: https://edition.cnn.com/site
User-agent: *
Allow: /partners/ipad/live-video.json
Disallow: /*.jsx$
Disallow: /ads/
Disallow: /aol/
Disallow: /audio/
Disallow: /beta/
Disallow: /browsers/
Disallow: /cl/
Disallow: /cnews/
Disallow: /cnn_adspaces
Disallow: /cnnbeta/
Disallow: /cnnintl_adspaces
Disallow: /development
Disallow: /editionssi
Disallow: /help/cnnx.html
Disallow: /NewsPass
Disallow: /NOKIA
Disallow: /partners/
Disallow: /pipeline/
Disallow: /pointroll/
Disallow: /POLLSERVER/
Disallow: /pr/
Disallow: /privacy
Disallow: /PV/
Disallow: /Quickcast/
Disallow: /quickcast/
Disallow: /QUICKNEWS/
Disallow: /search/
Disallow: /terms
Disallow: /test/
Disallow: /virtual/
Disallow: /WEB-INF/
Disallow: /web.projects/
Disallow: /webview/
```

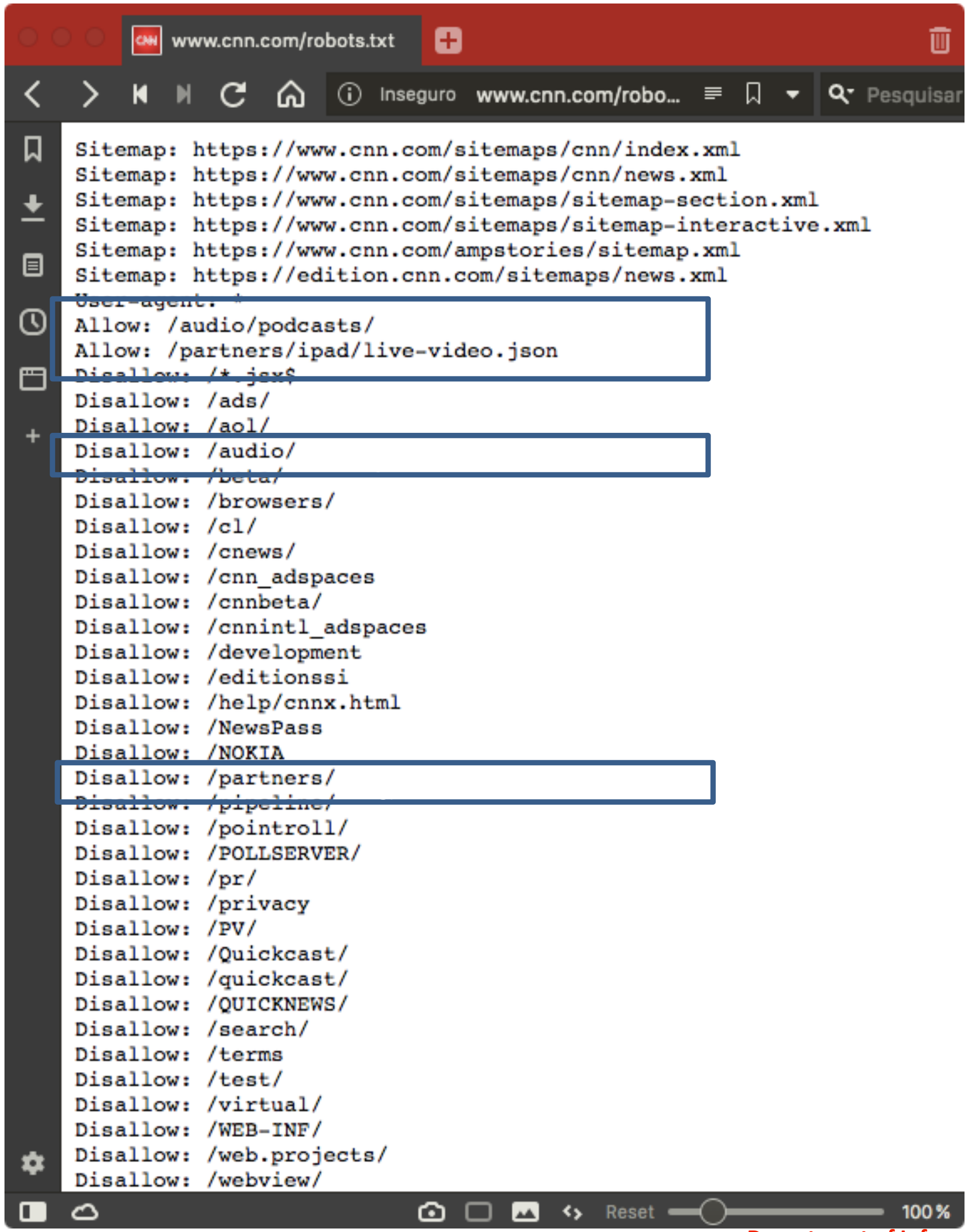
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:image="http://www.google.com/schemas/sitemap-image/1.1">
  ▼<url>
    ▼<loc>
       https://www.cnn.com/2012/01/25/world/gallery/wus-us-
       mia/index.html
     </loc>
     <lastmod>2012-01-25T18:24:37Z</lastmod>
     <changefreq>daily</changefreq>
     <priority>0.5</priority>
    ▼<image:image>
      ▼<image:loc>
         https://i2.cdn.turner.com/cnnnext/dam/assets/120111071718-
         jpac1-plane-story-top.jpg
       </image:loc>
      ▼<image:caption>
         Gunnery Sgt. Bryon Bebout with the wreckage of a B-24
         Liberator during excavation operations in Papua New Guinea.
         The JPAC recovery team were searching for nine Americans
         that remain unaccounted-for from World War II.
       </image:caption>
     </image:image>
   </url>
  ▼<url>
    ▼<loc>
       https://www.cnn.com/2012/01/18/living/gallery/cindy-
       costa/index.html
     </loc>
     <lastmod>2012-01-18T17:21:49Z</lastmod>
     <changefreq>daily</changefreq>
     <priority>0.5</priority>
    ▼<image:image>
      ▼<image:loc>
         https://i2.cdn.turner.com/cnnnext/dam/assets/120117082827-
         cindy-costa-01-story-top.jpg
       </image:loc>
      ▼<image:caption>
```

```
have any style information associated with it.
/www.sitemaps.org/schemas/sitemap/0.9">
/sitemaps/video-2020-01.xml</loc>
3:06Z</lastmod>
/sitemaps/article-2020-01.xml</loc>
0:01Z</lastmod>
m/sitemaps/article-2020-01.xml
8:27Z</lastmod>
/sitemaps/special-2020-01.xml</loc>
5:37Z</lastmod>
/sitemaps/show-2020-01.xml</loc>
3:54Z</lastmod>
/sitemaps/profile-2020-01.xml</loc>
6:12Z</lastmod>
/sitemaps/gallery-2020-01.xml</loc>
5:13Z</lastmod>
m/sitemaps/gallery-2020-01.xml
1:02Z</lastmod>
```

Information to crawlers

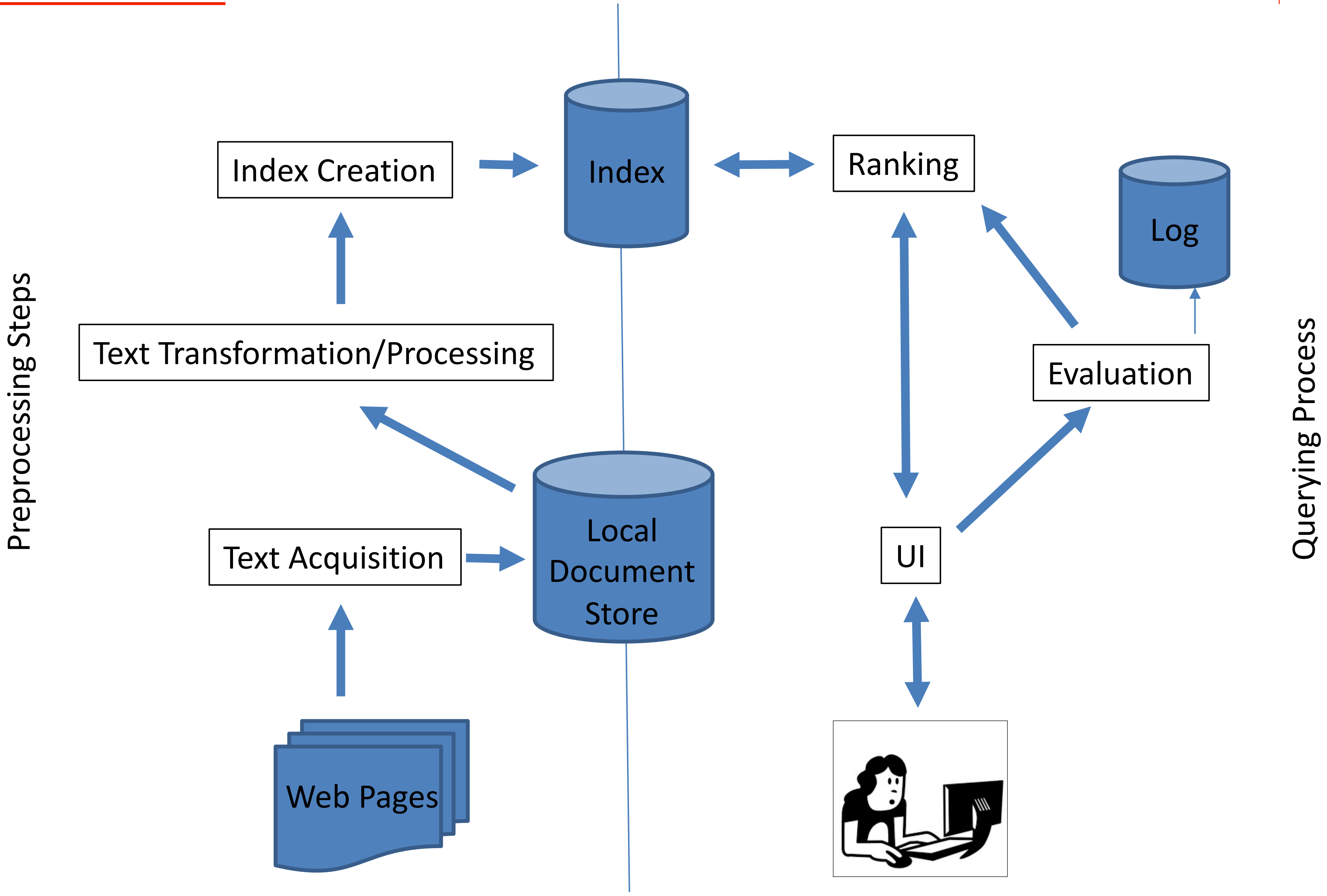
http://www.cnn.com/robots.txt



```
Sitemap: https://www.cnn.com/sitemaps/cnn/index.xml
Sitemap: https://www.cnn.com/sitemaps/cnn/news.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-section.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-interactive.xml
Sitemap: https://www.cnn.com/ampstories/sitemap.xml
Sitemap: https://edition.cnn.com/sitemaps/news.xml
User-agent: *
Allow: /audio/podcasts/
Allow: /partners/ipad/live-video.json
Disallow: /*.jsx$
Disallow: /ads/
Disallow: /aol/
Disallow: /audio/
Disallow: /beta/
Disallow: /browsers/
Disallow: /cl/
Disallow: /cnews/
Disallow: /cnn_adspaces
Disallow: /cnnbeta/
Disallow: /cnnintl_adspaces
Disallow: /development
Disallow: /editionssi
Disallow: /help/cnnx.html
Disallow: /NewsPass
Disallow: /NOKIA
Disallow: /partners/
Disallow: /pipeline/
Disallow: /pointroll/
Disallow: /POLLSERVER/
Disallow: /pr/
Disallow: /privacy
Disallow: /PV/
Disallow: /Quickcast/
Disallow: /quickcast/
Disallow: /QUICKNEWS/
Disallow: /search/
Disallow: /terms
Disallow: /test/
Disallow: /virtual/
Disallow: /WEB-INF/
Disallow: /web.projects/
Disallow: /webview/
```

- **You can allow a specific resource from a disallowed resource**
  - For instance: *you can disallow to crawl some folder, but allow the bot to crawl a resource that contains a list of links that you want to be indexed.*

Architecture



Pseudo Code

```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

“Basic algorithm” is...

- Theoretically correct

“Basic algorithm” is...

- Theoretically correct
- **Seriously lacking to use in practice**

“Basic algorithm” is...

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers

“Basic algorithm” is…

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers
  2. Very slow
     - 1 page at a time

“Basic algorithm” is...

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers
  2. Very slow
     - 1 page at a time
  3. Will get caught in traps **and** infinite sequences

“Basic algorithm” is...

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers
  2. Very slow
     - 1 page at a time
  3. Will get caught in traps and infinite sequences
  4. Will fetch duplicates without noticing

“Basic algorithm” is...

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers
  2. Very slow
     - 1 page at a time
  3. Will get caught in traps and infinite sequences
  4. Will fetch duplicates without noticing
  5. Will bring in data noise

“Basic algorithm” is…

- Theoretically correct
- **Seriously lacking to use in practice**
  1. Will upset web admins (impolite)
     - It’s abusing the web servers
  2. Very slow
     - 1 page at a time
  3. Will get caught in traps and infinite sequences
  4. Will fetch duplicates without noticing
  5. Will bring in data noise
  6. Will miss content due to client-side scripting

1. Politeness

- Avoid hitting any site too often
  - Sites are for people, not for bots
- Ignore politeness → Denial of service (DOS) attack
- **Be polite** → Use artificial delays

2. Performance (I)

- Back of the envelope calculation:
  - 1 page fetch = 500ms
  - How much time to crawl 1 million pages?
    - (it's worse than that... Unresponsive servers)
- Most of the time, the crawler thread is waiting for the network data
- Solution: multi-threaded or distributed crawling
  - Politeness is harder to control, but it is possible (e.g. different servers)

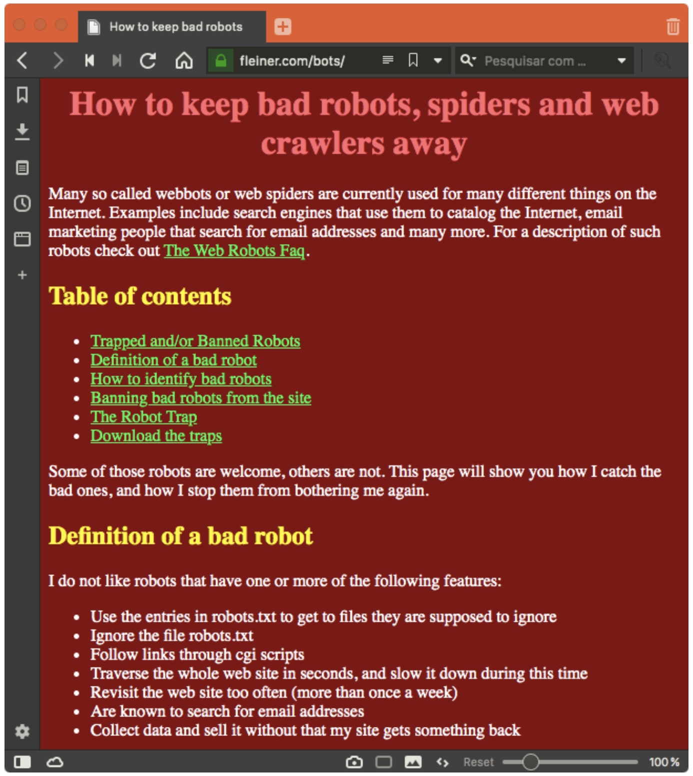
2. Performance (II)

- Domain Name lookups
  - Given a domain name, retrieve its IP address
    - www.ics.uci.edu -> 128.195.1.83
- Distributed set of servers
  - Latency can be high (2 secs is not unusual)
- Common implementations are blocking
  - One request at a time
  - Result is cached
- Back of the envelope calculation:
  - 1 DNS lookup ➔ 800ms
  - How much time to lookup the entire Web?

3. Crawler traps

- **May** trap the crawler on the site forever
  - Web server responds with ever changing URLs and content
    - Dynamic pages
  - Can be intentional or unintentional
    - E.g. the ICS calendar is a crawler trap
    - Some webadmins can create traps to penalize impolite crawlers

- See http://www.fleiner.com/bots/
    - E.g. very large documents, disalowed in robots.txt, created to consume crawler resources or event to break poorly designed parsers of crawlers that ignore robots.txt



How to keep bad robots, spiders and web crawlers away

Many so called webbots or web spiders are currently used for many different things on the Internet. Examples include search engines that use them to catalog the Internet, email marketing people that search for email addresses and many more. For a description of such robots check out The Web Robots Faq.

Table of contents

Some of those robots are welcome, others are not. This page will show you how I catch the bad ones, and how I stop them from bothering me again.

Definition of a bad robot

I do not like robots that have one or more of the following features:

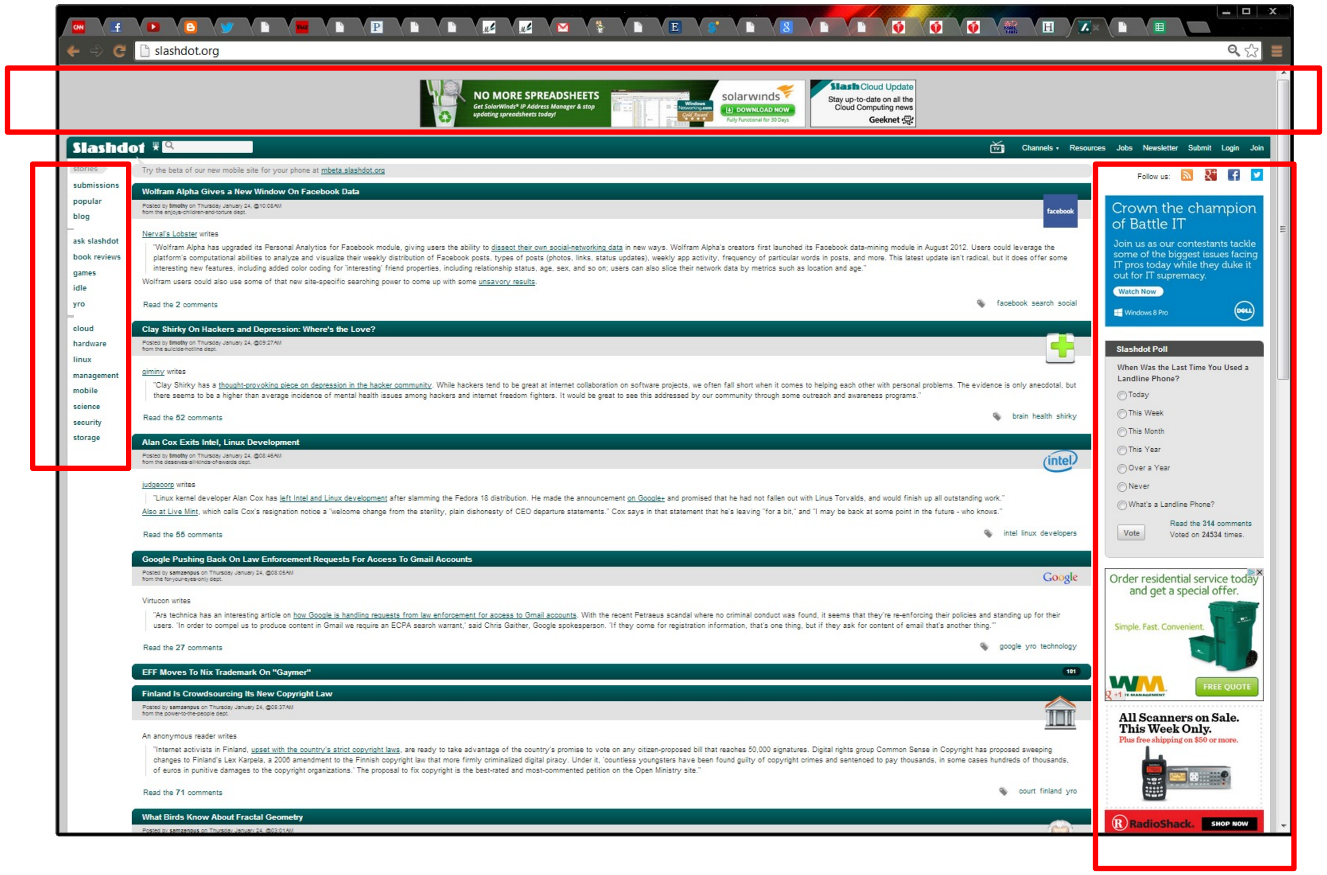
- Use the entries in robots.txt to get to files they are supposed to ignore
- Ignore the file robots.txt
- Follow links through cgi scripts
- Traverse the whole web site in seconds, and slow it down during this time
- Revisit the web site too often (more than once a week)
- Are known to search for email addresses
- Collect data and sell it without that my site gets something back

4. Duplicate Detection

- Exact and near duplication are widespread
  - Copies, mirror sites, versions, spam, plagiarism...
  - Studies: 30% of Web pages are [near-]duplicates of the other 70%
  - Little or no value (noise to the search engine and the user; you can show only one to the user and perhaps a “show similar” link)
- Detection
  - Detection of exact duplication is easy, but *exact duplication is rare*
    - Hashes, checksums
  - Detection of *near-duplicates* is harder
    - Page fingerprints

5. Data Noise

- Web pages have content not directly related to the page
  - Ads, templates, etc
  - Noise negatively impacts information retrieval


slashdot.org
NO MORE SPREADSHEETS
Get SolarWinds® IP Address Manager & stop updating spreadsheets today!
solarwinds
DOWNLOAD NOW
Fully Functional for 30 Days
SlashCloud Update
Stay up-to-date on all the Cloud Computing news
Geeknet
Slashdot
Channels
Resources
Jobs
Newsletter
Submit
Login
Join
stories
submissions
popular
blog
ask slashdot
book reviews
games
idle
yro
cloud
hardware
linux
management
mobile
science
security
storage
Follow us:
Crown the champion of Battle IT
Join us as our contestants tackle some of the biggest issues facing IT pros today while they duke it out for IT supremacy.
Watch Now
Windows 8 Pro
Order residential service today and get a special offer.
Simple. Fast. Convenient.
FREE QUOTE
All Scanners on Sale. This Week Only.
Plus free shipping on $50 or more.
RadioShack.
SHOP NOW

Try the beta of our new mobile site for your phone at mbeta.slashdot.org

Wolfram Alpha Gives a New Window On Facebook Data

Posted by timothy on Thursday January 24, @10:08AM
from the enjoys-children-and-torture dept.

Nerval's Lobster writes

"Wolfram Alpha has upgraded its Personal Analytics for Facebook module, giving users the ability to dissect their own social-networking data in new ways. Wolfram Alpha's creators first launched its Facebook data-mining module in August 2012. Users could leverage the platform's computational abilities to analyze and visualize their weekly distribution of Facebook posts, types of posts (photos, links, status updates), weekly app activity, frequency of particular words in posts, and more. This latest update isn't radical, but it does offer some interesting new features, including added color coding for 'interesting' friend properties, including relationship status, age, sex, and so on; users can also slice their network data by metrics such as location and age."

Wolfram users could also use some of that new site-specific searching power to come up with some unsavory results.

Read the 2 comments — facebook search social

Clay Shirky On Hackers and Depression: Where's the Love?

Posted by timothy on Thursday January 24, @09:27AM
from the suicide-hotline dept.

giminy writes

"Clay Shirky has a thought-provoking piece on depression in the hacker community. While hackers tend to be great at internet collaboration on software projects, we often fall short when it comes to helping each other with personal problems. The evidence is only anecdotal, but there seems to be a higher than average incidence of mental health issues among hackers and internet freedom fighters. It would be great to see this addressed by our community through some outreach and awareness programs."

Read the 52 comments — brain health shirky

Alan Cox Exits Intel, Linux Development

Posted by timothy on Thursday January 24, @08:46AM
from the deserves-all-kinds-of-awards dept.

judgecorp writes

"Linux kernel developer Alan Cox has left Intel and Linux development after slamming the Fedora 18 distribution. He made the announcement on Google+ and promised that he had not fallen out with Linus Torvalds, and would finish up all outstanding work."

Also at Live Mint, which calls Cox's resignation notice a "welcome change from the sterility, plain dishonesty of CEO departure statements." Cox says in that statement that he's leaving "for a bit," and "I may be back at some point in the future - who knows."

Read the 55 comments — intel linux developers

Google Pushing Back On Law Enforcement Requests For Access To Gmail Accounts

Posted by samzenpus on Thursday January 24, @08:05AM
from the for-your-eyes-only dept.

Virtucon writes

"Ars technica has an interesting article on how Google is handling requests from law enforcement for access to Gmail accounts. With the recent Petraeus scandal where no criminal conduct was found, it seems that they're re-enforcing their policies and standing up for their users. 'In order to compel us to produce content in Gmail we require an ECPA search warrant,' said Chris Gaither, Google spokesperson. 'If they come for registration information, that's one thing, but if they ask for content of email that's another thing.'"

Read the 27 comments — google yro technology

EFF Moves To Nix Trademark On "Gaymer" 101

Finland Is Crowdsourcing Its New Copyright Law

Posted by samzenpus on Thursday January 24, @05:37AM
from the power-to-the-people dept.

An anonymous reader writes

"Internet activists in Finland, upset with the country's strict copyright laws, are ready to take advantage of the country's promise to vote on any citizen-proposed bill that reaches 50,000 signatures. Digital rights group Common Sense in Copyright has proposed sweeping changes to Finland's Lex Karpela, a 2006 amendment to the Finnish copyright law that more firmly criminalized digital piracy. Under it, 'countless youngsters have been found guilty of copyright crimes and sentenced to pay thousands, in some cases hundreds of thousands, of euros in punitive damages to the copyright organizations.' The proposal to fix copyright is the best-rated and most-commented petition on the Open Ministry site."

Read the 71 comments — court finland yro

What Birds Know About Fractal Geometry

Posted by samzenpus on Thursday January 24, @03:01AM

Slashdot Poll

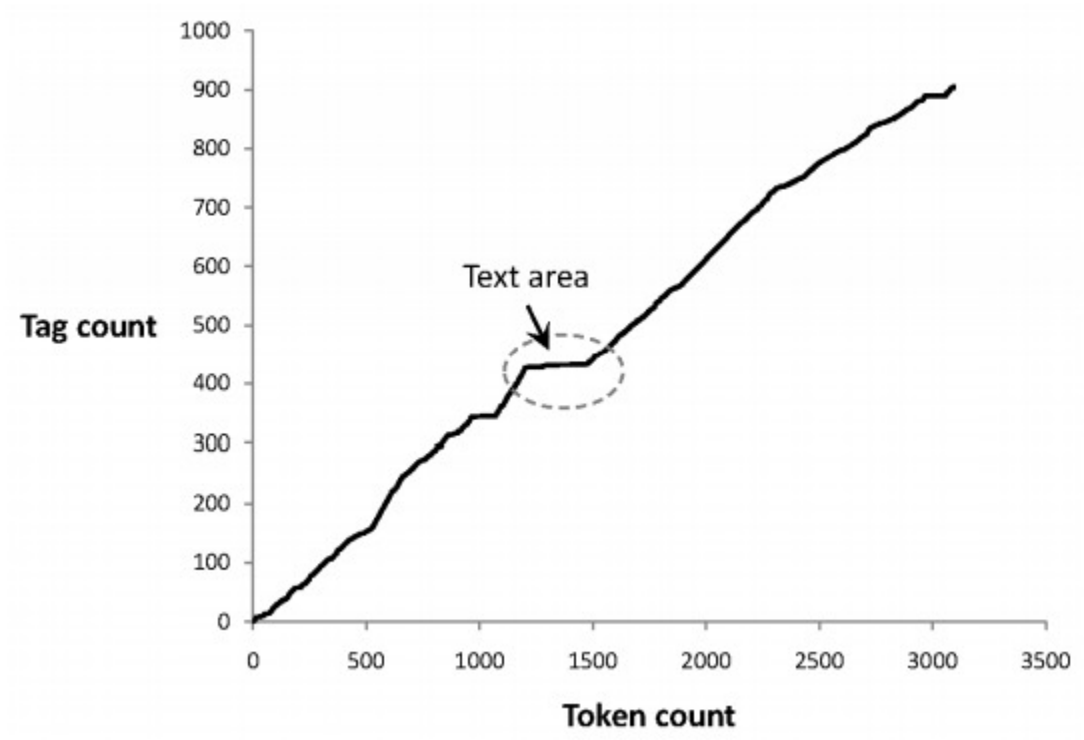
When Was the Last Time You Used a Landline Phone?

- Today
- This Week
- This Month
- This Year
- Over a Year
- Never
- What's a Landline Phone?

Vote — Read the 314 comments — Voted on 24534 times.

5. Data Noise : Finding Content Blocks

- Technique 1: Cumulative distribution of tags
  - Document slope curve (e.g. Finn, Kushmerick & Smyth, 2001)



- Other techniques in literature

6. Client-Side Scripting

- Modern web sites are heavily scripted (JavaScript, TypeScript)
  - Content behind XMLHttpRequests

6. Client-Side Scripting

- Modern web sites are heavily scripted (JavaScript, TypeScript)
  - Content behind XMLHttpRequests
- To get to that content crawlers must interact with the scripts
  - Hard thing to do: user interaction emulation (e.g. Selenium)
  - Most crawlers will not do it and the content will be never indexed

The Deep Web

- Places where crawlers rarely go...
  - Content behind login forms
  - Content behind JavaScript/TypeScript
  - Sites that aren't linked from anywhere

- It is estimated that the deep web is 400-500x larger than the shallow web [http://dx.doi.org/10.3998/3336451.0007.104]