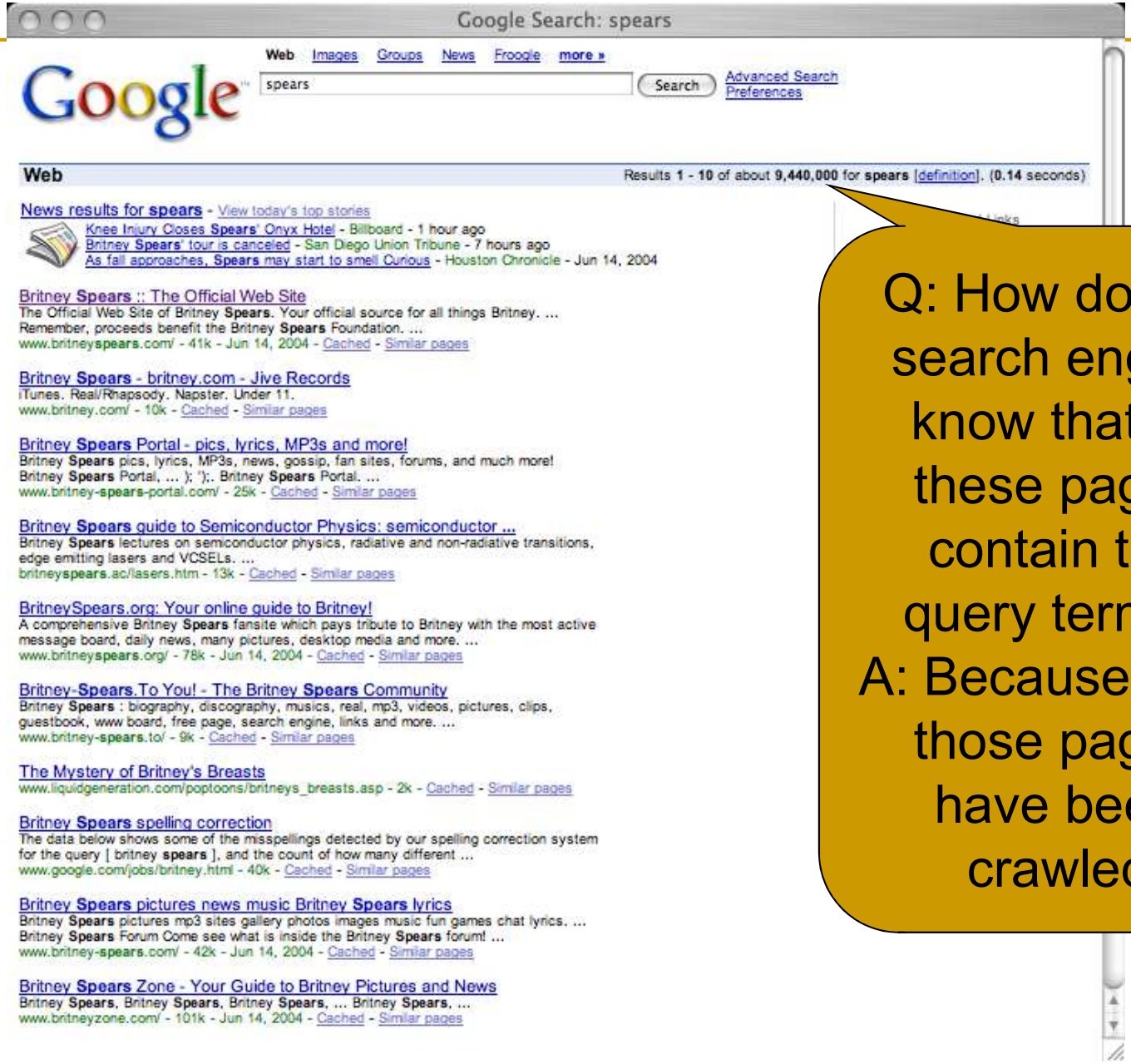
Ch. 8: Web Crawling

Outline

- Motivation and taxonomy of crawlers
- Basic crawlers and implementation issues
- Universal crawlers
- Preferential (focused and topical) crawlers
- Crawler ethics and conflicts


Google Search: spears
Google
Web Images Groups News Froogle more »
spears
Search
Advanced Search
Preferences
Web
Results 1 - 10 of about 9,440,000 for spears [definition]. (0.14 seconds)
News results for spears - View today's top stories
Knee Injury Closes Spears' Onyx Hotel - Billboard - 1 hour ago
Britney Spears' tour is canceled - San Diego Union Tribune - 7 hours ago
As fall approaches, Spears may start to smell Curious - Houston Chronicle - Jun 14, 2004
Britney Spears :: The Official Web Site
The Official Web Site of Britney Spears. Your official source for all things Britney. ...
Remember, proceeds benefit the Britney Spears Foundation. ...
www.britneyspears.com/ - 41k - Jun 14, 2004 - Cached - Similar pages
Britney Spears - britney.com - Jive Records
iTunes. Real/Rhapsody. Napster. Under 11.
www.britney.com/ - 10k - Cached - Similar pages
Britney Spears Portal - pics, lyrics, MP3s and more!
Britney Spears pics, lyrics, MP3s, news, gossip, fan sites, forums, and much more!
Britney Spears Portal, ...); ');. Britney Spears Portal. ...
www.britney-spears-portal.com/ - 25k - Cached - Similar pages
Britney Spears guide to Semiconductor Physics: semiconductor ...
Britney Spears lectures on semiconductor physics, radiative and non-radiative transitions, edge emitting lasers and VCSELs. ...
britneyspears.ac/lasers.htm - 13k - Cached - Similar pages
BritneySpears.org: Your online guide to Britney!
A comprehensive Britney Spears fansite which pays tribute to Britney with the most active message board, daily news, many pictures, desktop media and more. ...
www.britneyspears.org/ - 78k - Jun 14, 2004 - Cached - Similar pages
Britney-Spears.To You! - The Britney Spears Community
Britney Spears : biography, discography, musics, real, mp3, videos, pictures, clips, guestbook, www board, free page, search engine, links and more. ...
www.britney-spears.to/ - 9k - Cached - Similar pages
The Mystery of Britney's Breasts
www.liquidgeneration.com/poptoons/britneys_breasts.asp - 2k - Cached - Similar pages
Britney Spears spelling correction
The data below shows some of the misspellings detected by our spelling correction system for the query [britney spears], and the count of how many different ...
www.google.com/jobs/britney.html - 40k - Cached - Similar pages
Britney Spears pictures news music Britney Spears lyrics
Britney Spears pictures mp3 sites gallery photos images music fun games chat lyrics. ...
Britney Spears Forum Come see what is inside the Britney Spears forum! ...
www.britney-spears.com/ - 42k - Jun 14, 2004 - Cached - Similar pages
Britney Spears Zone - Your Guide to Britney Pictures and News
Britney Spears, Britney Spears, Britney Spears, ... Britney Spears, ...
www.britneyzone.com/ - 101k - Jun 14, 2004 - Cached - Similar pages
Q: How does a search engine know that all these pages contain the query terms?
A: Because all of those pages have been crawled

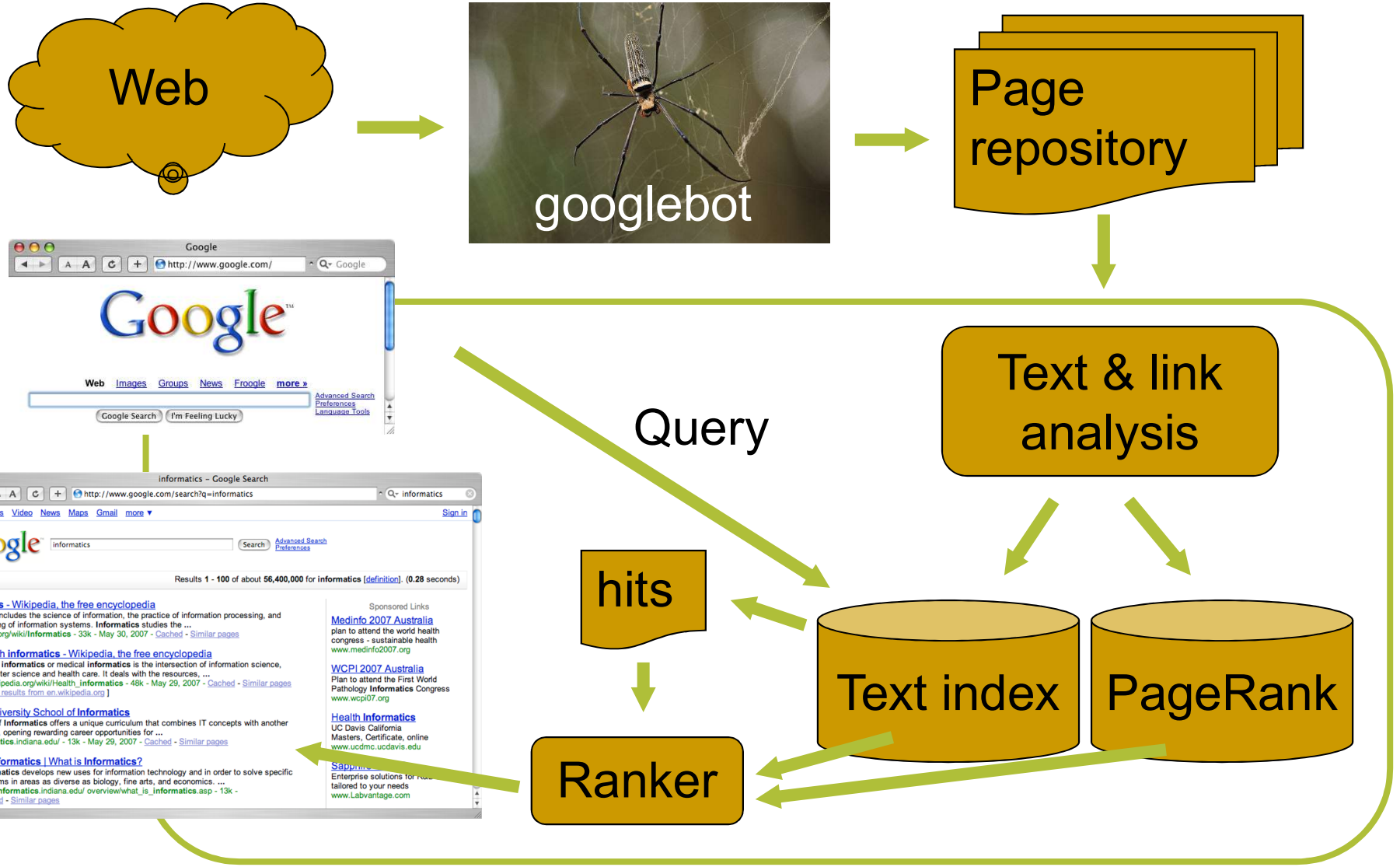
Many names

- Crawler
- Spider
- Robot (or bot)
- Web agent
- Wanderer, worm, …
- And famous instances: googlebot, scooter, slurp, msnbot, …

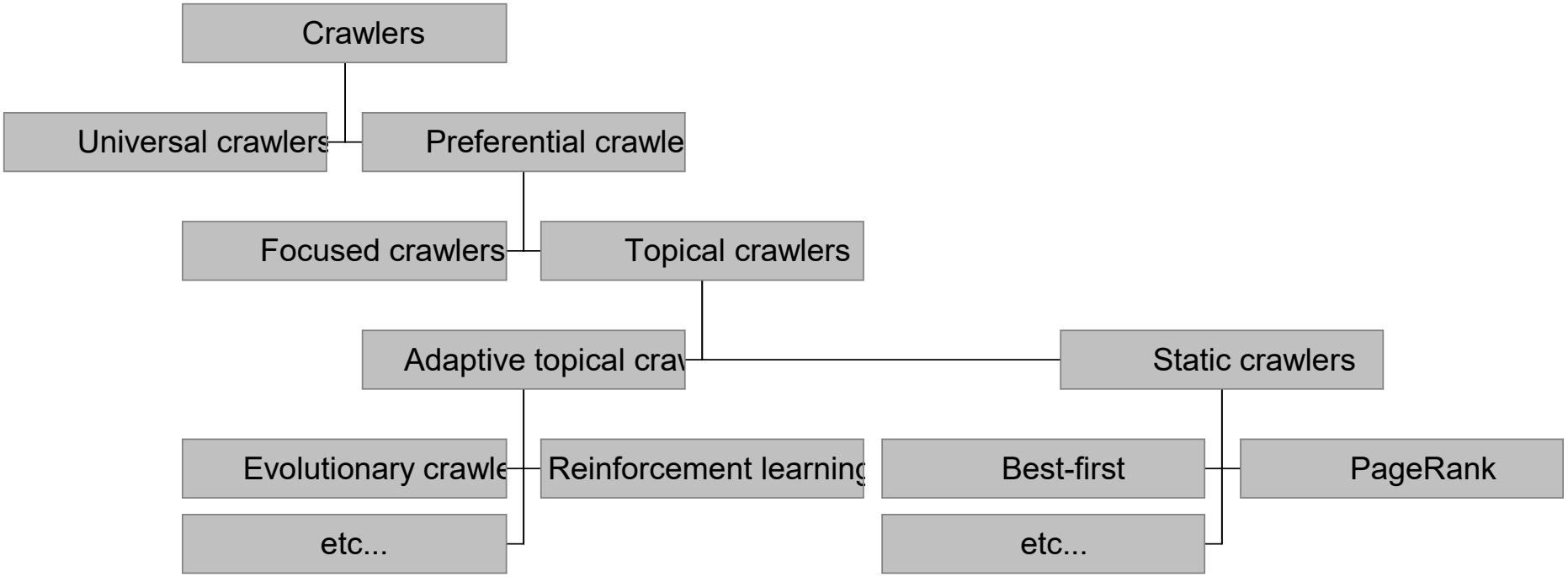
Motivation for crawlers

- Support universal search engines (Google, Yahoo, MSN/Windows Live, Ask, etc.)
- Vertical (specialized) search engines, e.g. news, shopping, papers, recipes, reviews, etc.
- Business intelligence: keep track of potential competitors, partners
- Monitor Web sites of interest
- Evil: harvest emails for spamming, phishing…
- … Can you think of some others?…

A crawler within a search engine



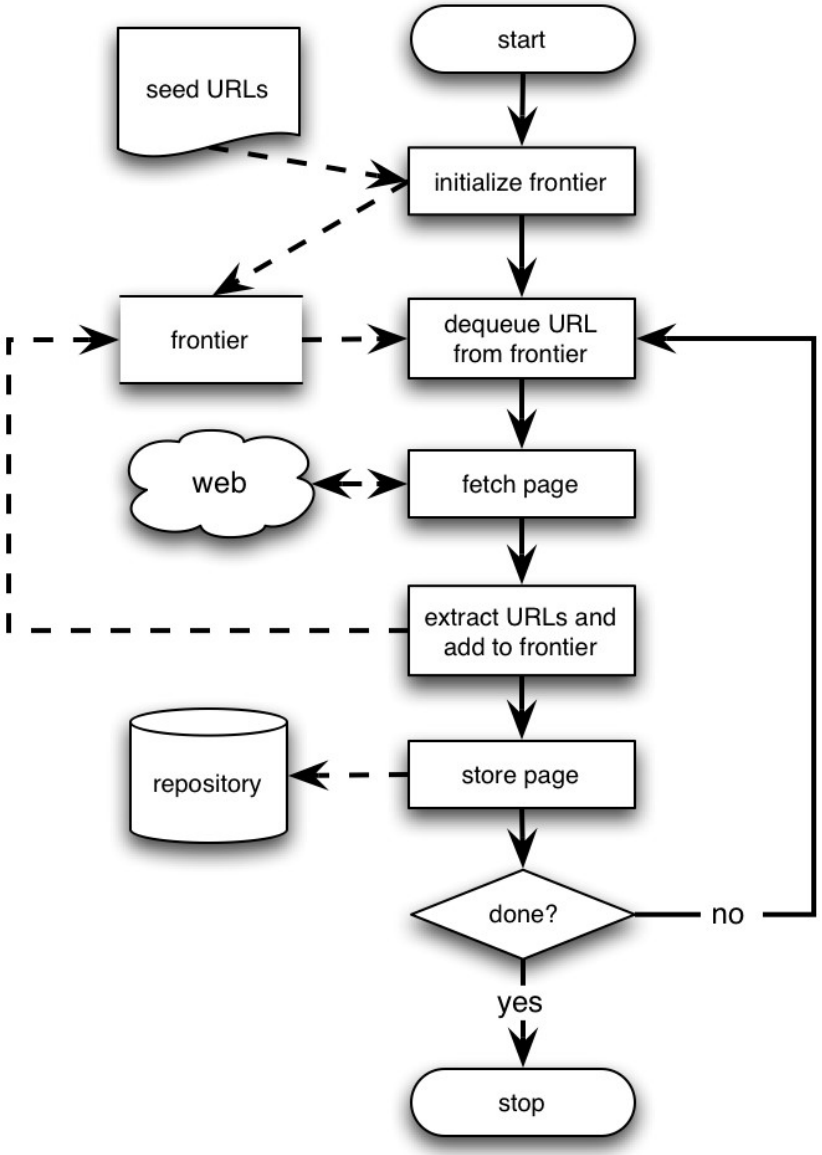
One taxonomy of crawlers



- Many other criteria could be used:
  - Incremental, Interactive, Concurrent, Etc.

Outline

- Motivation and taxonomy of crawlers
- Basic crawlers and implementation issues
- Universal crawlers
- Preferential (focused and topical) crawlers
- Crawler ethics and conflicts

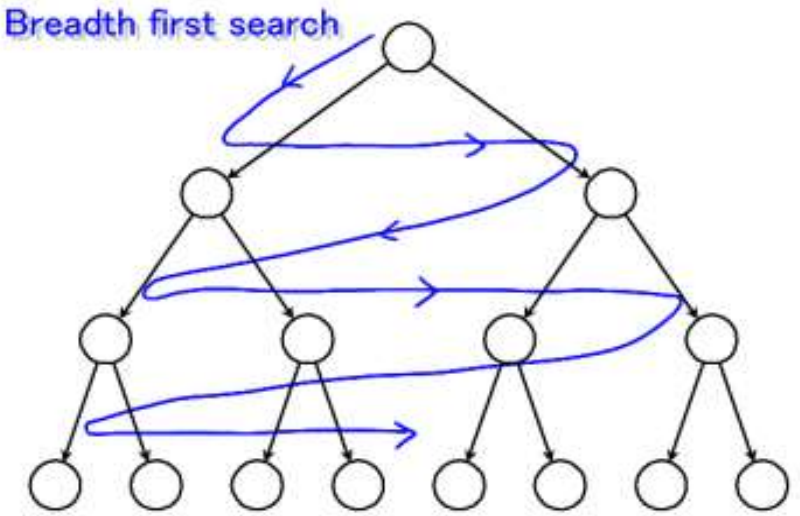


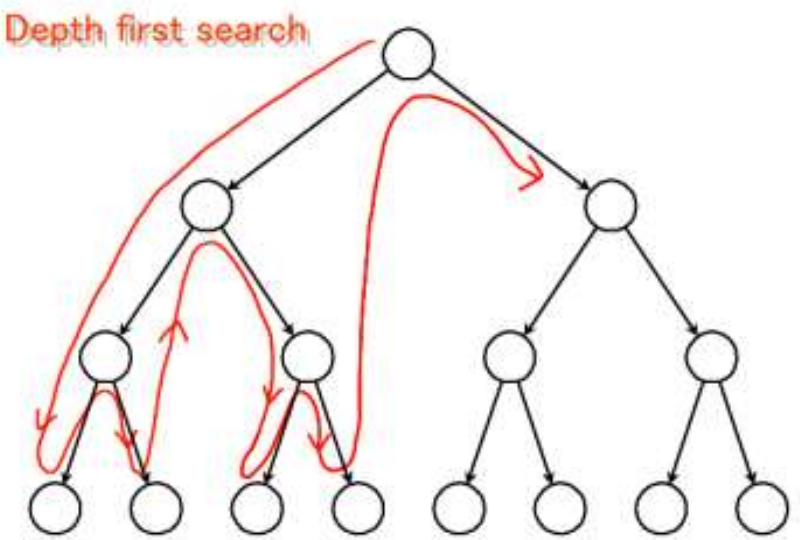
Basic crawlers

- This is a sequential crawler
- Seeds can be any list of starting URLs
- Order of page visits is determined by frontier data structure
- Stop criterion can be anything

Graph traversal (BFS or DFS?)

- Breadth First Search
  - Implemented with QUEUE (FIFO)
  - Finds pages along shortest paths
  - If we start with “good” pages, this keeps us close; maybe other good stuff…
- Depth First Search
  - Implemented with STACK (LIFO)
  - Wander away (“lost in cyberspace”)





Implementation issues

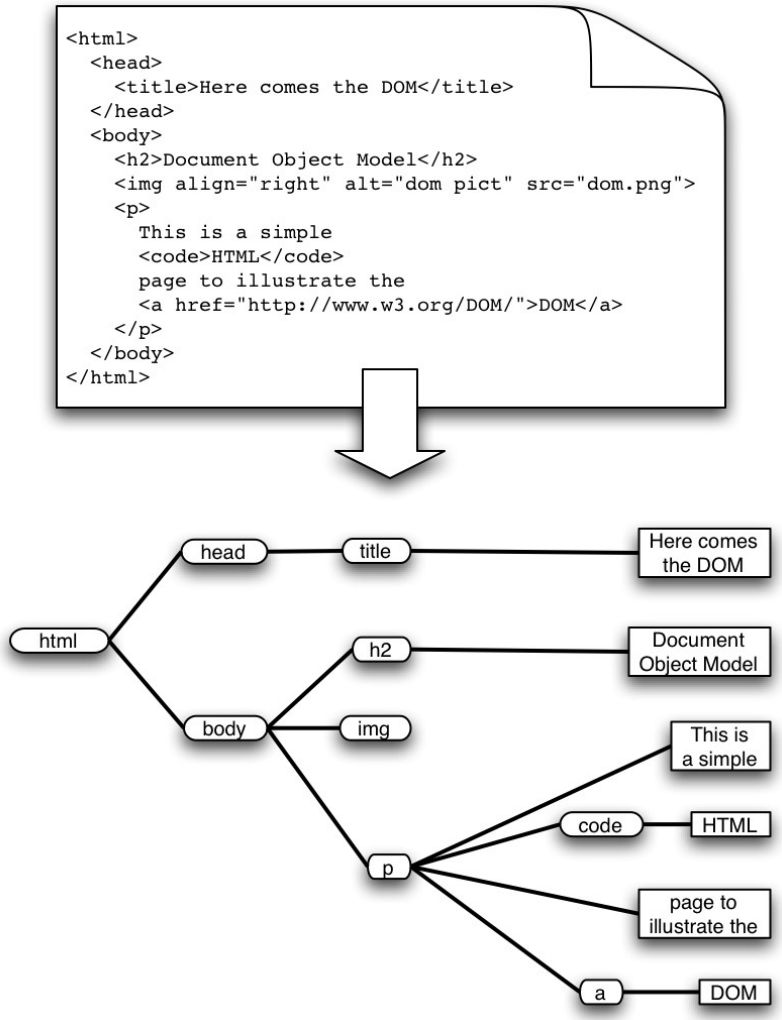
- Don’t want to fetch same page twice!
  - Keep lookup table (hash) of visited pages
  - What if not visited but in frontier already?
- The frontier grows very fast!
  - May need to prioritize for large crawls
- Fetcher must be robust!
  - Don’t crash if download fails
  - Timeout mechanism
- Determine file type to skip unwanted files
  - Can try using extensions, but not reliable
  - Can issue ‘HEAD’ HTTP commands to get Content-Type (MIME) headers, but overhead of extra Internet requests

More implementation issues

- Fetching
  - Get only the first 10-100 KB per page
  - Take care to detect and break redirection loops
  - Soft fail for timeout, server not responding, file not found, and other errors

More implementation issues: Parsing

- HTML has the structure of a DOM (Document Object Model) tree
- Unfortunately actual HTML is often incorrect in a strict syntactic sense
- Crawlers, like browsers, must be robust/forgiving
- Fortunately there are tools that can help
  - E.g. tidy.sourceforge.net
- Must pay attention to HTML entities and unicode in text
- What to do with a growing number of other formats?
  - Flash, SVG, RSS, AJAX…



More implementation issues

- Stop words
  - Noise words that do not carry meaning should be eliminated ("stopped") before they are indexed
  - E.g. in English: AND, THE, A, AT, OR, ON, FOR, etc…
  - Typically syntactic markers
  - Typically the most common terms
  - Typically kept in a negative dictionary
    - 10–1,000 elements
    - E.g. http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
  - Parser can detect these right away and disregard them

More implementation issues

Conflation and thesauri

- Idea: improve recall by merging words with same meaning

1. We want to ignore superficial morphological features, thus merge semantically similar tokens
   - {student, study, studying, studious} => studi
2. We can also conflate synonyms into a single form using a thesaurus
   - 30-50% smaller index
   - Doing this in both pages and queries allows to retrieve pages about ‘automobile’ when user asks for ‘car’
   - Thesaurus can be implemented as a hash table

More implementation issues

- Stemming
  - Morphological conflation based on rewrite rules
  - Language dependent!
  - Porter stemmer very popular for English
    - http://www.tartarus.org/~martin/PorterStemmer/
    - Context-sensitive grammar rules, eg:
      - "IES" except ("EIES" or "AIES") --> "Y"
    - Versions in Perl, C, Java, Python, C#, Ruby, PHP, etc.
  - Porter has also developed Snowball, a language to create stemming algorithms in any language
    - http://snowball.tartarus.org/
    - Ex. Perl modules: Lingua::Stem and Lingua::Stem::Snowball

More implementation issues

- Static vs. dynamic pages
  - Is it worth trying to eliminate dynamic pages and only index static pages?
  - Examples:
    - https://www.youtube.com/
  - Why or why not? How can we tell if a page is dynamic? What about ‘spider traps’?
  - What do Google and other search engines do?

More implementation issues

- Relative vs. Absolute URLs
  - Crawler must translate relative URLs into absolute URLs
  - Need to obtain Base URL from HTTP header, or HTML Meta tag, or else current page path by default
  - Examples
    - **Base:** `http://www.cnn.com/linkto/`
    - **Relative URL:** `intl.html`
    - **Absolute URL:** `http://www.cnn.com/linkto/intl.html`
    - **Relative URL:** `/US/`
    - **Absolute URL:** `http://www.cnn.com/US/`

More implementation issues

- URL canonicalization
  - All of these:
    - http://www.cnn.com/TECH
    - http://WWW.CNN.COM/TECH/
    - http://www.cnn.com:80/TECH/
    - http://www.cnn.com/bogus/../TECH/
  - Are really equivalent to this canonical form:
    - **http://www.cnn.com/TECH/**
  - In order to avoid duplication, the crawler must transform all URLs into canonical form
  - Definition of “canonical” is arbitrary, e.g.:
    - Could always include port
    - Or only include port when not default :80

More on Canonical URLs

- Some transformation are trivial, for example:
  - ✗ http://informatics.indiana.edu
  - ✓ http://informatics.indiana.edu/
  - ✗ http://informatics.indiana.edu/index.html#fragment
  - ✓ http://informatics.indiana.edu/index.html
  - ✗ http://informatics.indiana.edu/dir1/./../dir2/
  - ✓ http://informatics.indiana.edu/dir2/
  - ✗ http://informatics.indiana.edu/%7Efil/
  - ✓ http://informatics.indiana.edu/~fil/
  - ✗ http://INFORMATICS.INDIANA.EDU/fil/
  - ✓ http://informatics.indiana.edu/fil/

More on Canonical URLs

Other transformations require heuristic assumption about the intentions of the author or configuration of the Web server:

1. Removing default file name
   - ✓ `http://informatics.indiana.edu/fil/index.html`
   - × `http://informatics.indiana.edu/fil/`
   - This is reasonable in general but would be wrong in this case because the default happens to be ‘default.asp’ instead of ‘index.html’
2. Trailing directory
   - × `http://informatics.indiana.edu/fil`
   - ✓ `http://informatics.indiana.edu/fil/`
   - This is correct in this case but how can we be sure in general that there isn’t a file named ‘fil’ in the root dir?

More implementation issues

- Spider traps
  - Misleading sites: indefinite number of pages dynamically generated by CGI scripts
  - Paths of arbitrary depth created using soft directory links and path rewriting features in HTTP server
  - Only heuristic defensive measures:
    - Check URL length; assume spider trap above some threshold, for example 128 characters
    - Watch for sites with very large number of URLs
    - Eliminate URLs with non-textual data types
    - May disable crawling of dynamic pages, if can detect

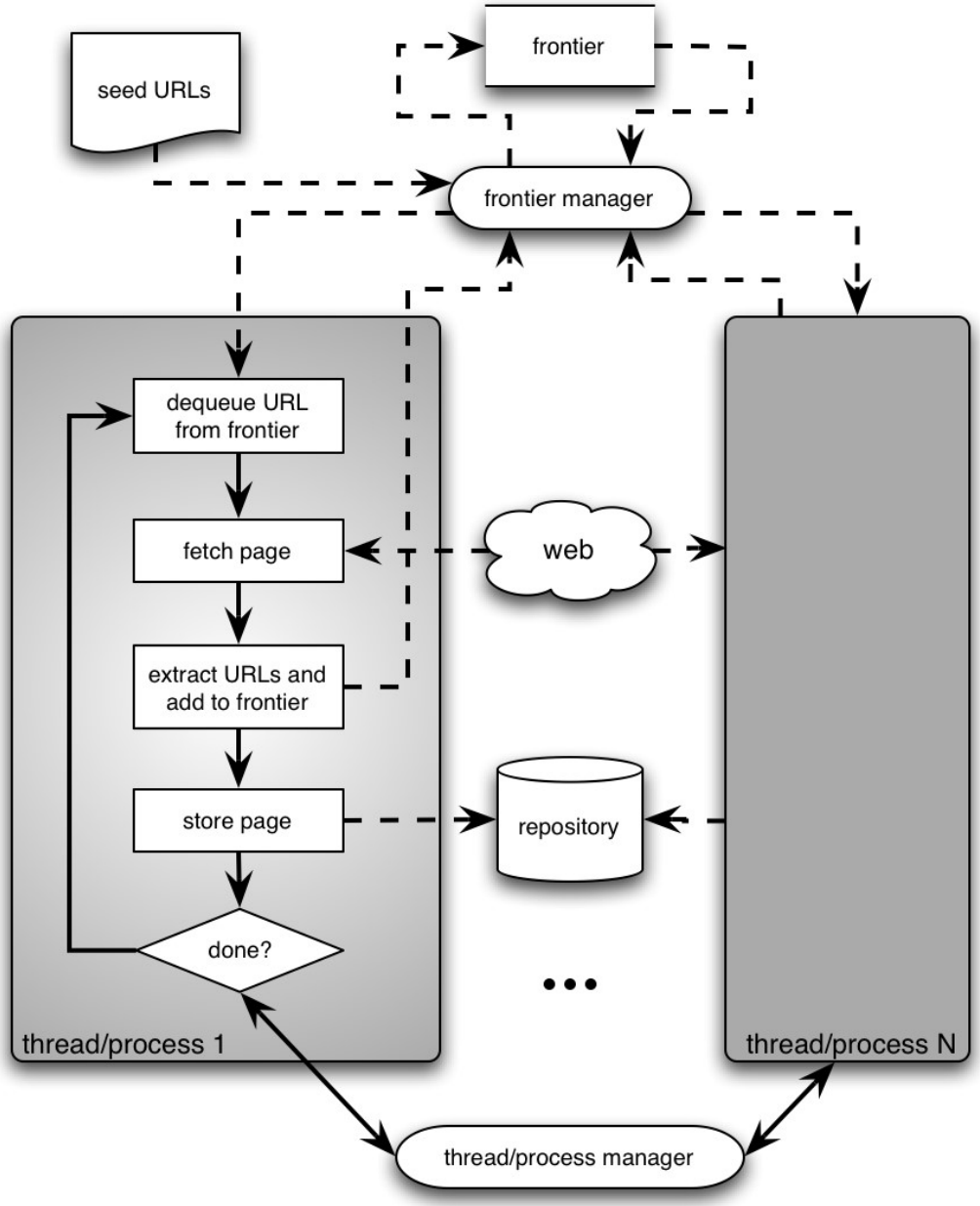
More implementation issues

- Page repository
  - Naïve: store each page as a separate file
    - Can map URL to unique filename using a hashing function, e.g. MD5
    - This generates a huge number of files, which is inefficient from the storage perspective
  - Better: combine many pages into a single large file, using some XML markup to separate and identify them
    - Must map URL to {filename, page_id}
  - Database options
    - Any RDBMS -- large overhead
    - Light-weight, embedded databases such as Berkeley DB

Concurrency

- A crawler incurs several delays:
  - Resolving the host name in the URL to an IP address using DNS
  - Connecting a socket to the server and sending the request
  - Receiving the requested page in response
- Solution: Overlap the above delays by fetching many pages concurrently

Architecture of a concurrent crawler



Concurrent crawlers

- Can use multi-processing or multi-threading
- Each process or thread works like a sequential crawler, except they share data structures: frontier and repository
- Shared data structures must be synchronized (locked for concurrent writes)
- Speedup of factor of 5-10 are easy this way

Outline

- Motivation and taxonomy of crawlers
- Basic crawlers and implementation issues
- Universal crawlers
- Preferential (focused and topical) crawlers
- Crawler ethics and conflicts

Universal crawlers

- Support universal search engines
- Large-scale
- Huge cost (network bandwidth) of crawl is amortized over many queries from users
- Incremental updates to existing index and other data repositories

Large-scale universal crawlers

- Two major issues:

1. Performance
   - Need to scale up to billions of pages
2. Policy
   - Need to trade-off coverage, freshness, and bias (e.g. toward “important” pages)

Universal crawlers: Policy

- Coverage
  - New pages get added all the time
  - Can the crawler find every page?
- Freshness
  - Pages change over time, get removed, etc.
  - How frequently can a crawler revisit ?
- Trade-off!
  - Focus on most “important” pages (crawler bias)?
  - “Importance” is subjective

Maintaining a "fresh" collection

- Universal crawlers are never "done"
- High variance in rate and amount of page changes
- HTTP headers are notoriously unreliable
  - Last-modified
  - Expires
- Solution
  - Estimate the probability that a previously visited page has changed in the meanwhile
  - Prioritize by this probability estimate

Do we need to crawl the entire Web?

- If we cover too much, it will get stale
- There is an abundance of pages in the Web
- For PageRank, pages with very low prestige are largely useless
- What is the goal?
  - General search engines: pages with high prestige
  - News portals: pages that change often
  - Vertical portals: pages on some topic
- What are appropriate priority measures in these cases? Approximations?