a tutorial on crawling tools

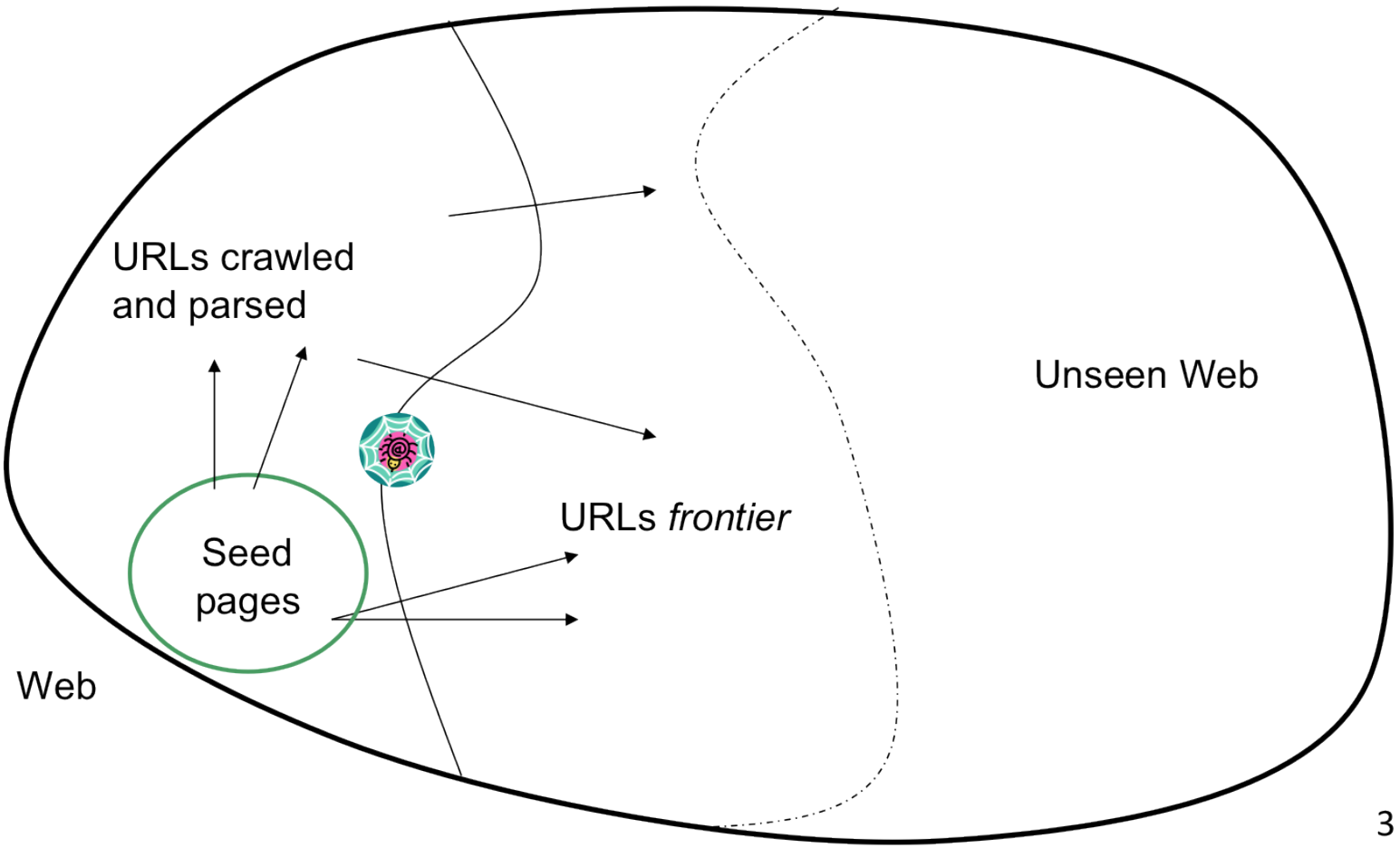
jianguo lu

October 27, 2024

Outline

type of crawlers

- classify crawling according to the type of the web:
  - surface web crawler: obtain web pages by following hyperlinks
  - deep web crawler
  - programmable web apis
- classify crawling according to the content:
  - general purpose, e.g., google. archive.com.
  - focused crawlers, e.g.,
    - academic crawlers (google scholar)
    - social networks (twitter, weibo)
- Surface web and deep web are intertwined
  - most large web sites provide both surface web and deep web (e.g., google scholar, twitter)


URLs crawled
and parsed
Unseen Web
URLs *frontier*
Seed
pages
Web

3

Tools for surface web downloading/crawling

- Command line
  - wget (www get), preinstalled in ubuntu(our cs machines)
  - curl (crawl url), OSX preinstalled
- Simple crawling apis
  - Java: crawler4j in java: http://code.google.com/p/crawler4j/
  - Python: scrapy: http://scrapy.org/
- Large scale scrawling
  - Heritrix, crawler for archive.org.
  - nutch

Starting example: get a webpage (in java)

```java
import java.net.*;
import java.io.*;
public class URLReader {
    public static void main(String[] args) throws Exception {

        URL oracle = new URL("http://www.oracle.com/");
        BufferedReader in = new BufferedReader(
        new InputStreamReader(oracle.openStream()));

        String inputLine;
        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);
        in.close();
    }
}
```

Every language can do the similar thing. e.g., in Matlab,

```
urlwrite(URL, filename)
```

limitations

- how to analyze the page to get other urls
- how to control the process
  - how deep to crawl
  - how often to send the request
  - ...

limitations

- how to analyze the page to get other urls
- how to control the process
  - how deep to crawl
  - how often to send the request
  - ...

wget

- stands for www get.
- developed in 1996
- preinstalled on most linux-like machines
- example to download a single file:

```
$ wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
Saving to: ‘strx25-0.9.2.1.tar.bz2.1’
31\% [=================> 1,213,592   68.2K/s  eta 34s
```

get more pages

- Get a single page
  - `wget http://www.example.com/index.html`
- Support http, ftp etc., e.g.
  - `wget ftp://ftp.gnu.org/pub/gnu/wget/wget-latest.tar.gz`
- More complex usage includes automatic download of multiple URLs into a directory hierarchy.
  - `wget -e robots=off -r -l1 --no-parent -A.gif ftp://www.example.com/dir/`
- Wikileaks was downloaded using one single command

Recursive retrieval using -r

- `-r -l1` Sets the depth level for recursion to 1. This means that wget will only download files in the specified directory (/dir/) without descending into any subdirectories.
- Setting a higher level, like -l2, would allow it to download files from subdirectories within /dir/.
- program begins following links from the website and downloading them too.
- `http://activehistory.ca/papers/` has a link to `http://activehistory.ca/papers/historypaper-9/`, so it will download that too if we use recursive retrieval.
- will also follow any other links: if there was a link to http://uwo.ca somewhere in that page, it would follow that and download it as well.
- By default, -r sends wget to a depth of five sites after the first one. This is following links, to a limit of five clicks after the first website.

not beyond last parent directory using --no-parent

```
wget -e robots=off -r -l1 --no-parent -A.gif
ftp://www.example.com/dir/
```

- The double-dash indicates the full-text of a command. All commands also have a short version, this could be initiated using -np.
- wget should follow links, but not beyond the last parent directory.
- won't go anywhere that is not part of the http://activehistory.ca/papers/hierarchy

how far you want to go

- The default: follow each link and carry on to a limit of five pages away from the first page.
- `wget -l 2`, which takes us to a depth of two web-pages.
- Note this is a lower-case L, not a number 1.

politeness

- `-w 10`
  - adds a ten second wait in between server requests.
  - you can shorten this, as ten seconds is quite long.
  - you can also use the parameter: –random-wait to let wget chose a random number of seconds to wait.
  - `wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com`
- `--limit-rate=20k`
  - limit the maximum download speed to 20kb/s.
  - Opinion varies on what a good limit rate is, but you are probably good up to about 200kb/s

some sites are protective

- if the robots.txt does not allow you to crawl anything
  - use robots=off
  - wget -r -p -e robots=off http://www.example.com

mask user agent

- if a web site checks a browser identity

```
$ wget -r -p -e  -U mozilla http://www.example.com
$ wget --user-agent="Mozilla/5.0 (X11; U; Linux i686;
en-US; rv:1.9.0.3) Gecko/2008092416 Firefox/3.0.3" URL-TO-
    DOWNLOAD
```

- The User-Agent string identifies the software (often a web browser) making the request.
- Web servers use the User-Agent string to identify the type of device, operating system, and browser being used.
- Some websites will only serve content or specific versions of the content to certain User-Agents, typically to prevent automated scraping or to display mobile-optimized content.

Increase Total Number of Retry Attempts

- By default wget retries 20 times to make the download successful.
- If the internet connection has problem, you may want to increase the number of tries

wget –tries=75 DOWNLOAD-URL

Download Multiple Files / URLs Using Wget -i

- First, store all the download files or URLs in a text file as:

```
download-file-list.txt
```

- Next, give the download-file-list.txt as argument to wget using -i option as shown below.

```
$ cat > download-file-list.txt
URL1
URL2
URL3
URL4
```

```
$ wget -i download-file-list.txt
```

Download Only Certain File Types Using wget -r -A

- You can use this under following situations:
  - Download all images from a website
  - Download all videos from a website
  - Download all PDF files from a website

```
$ wget -r -A.pdf http://url-to-webpage-with-pdfs/
```

download a directory

- task: download all the files under the papers directory of ActiveHistory.ca.
- `wget -r --no-parent -w 2 --limit-rate=20k http://activehistory.ca/papers/`
- Note that the trailing slash on the URL is critical
- if you omit it, wget will think that papers is a file rather than a directory.
- When it is done, you should have a directory labeled ActiveHistory.ca that contains the /papers/ sub-directory perfectly mirrored on your system.
- This directory will appear in the location that you ran the command from in your command line
- Links will be replaced with internal links to the other pages you've downloaded, so you can actually have a fully working ActiveHistory.ca site on your computer.

mirror a website using `-m`

- If you want to mirror an entire website, there is a built-in command to `wget`.
- This command means 'mirror', and is especially useful for backing up an entire website.
- it looks at the time stamps, and does not repeat the download if the file in the local system is recent.
- it supports infinite recursion (it will go as many layers into the site as necessary).
- The command for mirroring ActiveHistory.ca would be:
- `wget -m -w 2 --limit-rate=20k http://activehistory.ca`

download in the background using -b

- unattended download of large files

```
$ wget -b http://www.openss7.org/repos/tarballs/strx25
    -0.9.2.1.tar.bz2
Continuing in background, pid 1984.
Output will be written to ‘wget-log’.
```

Scrapy vs wget

- **wget**:
  - Primarily for downloading files, not data extraction.
  - Limited ability to handle JavaScript or AJAX.
  - Best suited for static, simple download tasks.
- **Scrapy and others**:
  - Designed specifically for web scraping.
  - Handles dynamic content and supports complex navigation.
  - Allows custom headers, cookies, and session management.

crawler4j

```
public class BasicCrawlController {
  public static void main(String[] args) throws Exception {
    String crawlStorageFolder = args[0];
    int numberOfCrawlers = Integer.parseInt(args[1]);
    CrawlConfig config = new CrawlConfig();
    config.setCrawlStorageFolder(crawlStorageFolder);
    config.setPolitenessDelay(1000);
    config.setMaxDepthOfCrawling(2);
    config.setMaxPagesToFetch(1000);
    config.setResumableCrawling(false);
    PageFetcher pageFetcher = new PageFetcher(config);
    RobotstxtConfig robotstxtConfig = new RobotstxtConfig();
    RobotstxtServer robotstxtServer = new RobotstxtServer(
        robotstxtConfig, pageFetcher);
    CrawlController controller = new CrawlController(config,
        pageFetcher, robotstxtServer);
    controller.addSeed("http://www.ics.uci.edu/");
    controller.start(BasicCrawler.class, numberOfCrawlers);
  }
}
```

general vs. special purpose crawlers

- many websites provide their own apis for crawling
  - facebook
  - google
  - twitter
  - nytimes
  - github
- each api has its own restrictions
- daily quota per day, per token, per IP

twitter

- there are several apis provided by twitter
  - streaming API
  - REST API
  - firehose
  - Mining the Social Web by Matthew A. Russell
  - 21 Recipes for Mining Twitter
- gnip.com provides api access to current and historical data

nutch overview

- Apache Nutch is an open source Web crawler written in Java.
- CommonCrawl is crawled by Nutch
  - contains almost all the web pages
  - billions of web pages
- Can find Webpage hyperlinks in an automated manner, reduce maintenance work
  - for example checking broken links
- Coupled with search engine (Solr)
  - create a copy of all the visited pages for searching over.