ECPR Methods Summer School:
Automated Collection of Web and Social Data

Pablo Barberá

School of International Relations
University of Southern California
`pablobarbera.com`

Networked Democracy Lab
`www.netdem.org`

Course website:
github.com/pablobarbera/ECPR-SC103

Scraping the web

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

These datasets are often provided in an unstructured format.

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

These datasets are often provided in an unstructured format.

Web scraping is the process of extracting this information automatically and transforming it into a structured dataset.

Scraping the web: two approaches

Two different approaches:

1. Screen scraping: extract data from source code of website, with html parser and/or regular expressions

Scraping the web: two approaches

Two different approaches:

1. Screen scraping: extract data from source code of website, with html parser and/or regular expressions
   - `rvest` package in R

Scraping the web: two approaches

Two different approaches:

1. Screen scraping: extract data from source code of website, with html parser and/or regular expressions
   - `rvest` package in R
2. Web APIs (application programming interfaces): a set of structured http requests that return JSON or XML data

Scraping the web: two approaches

Two different approaches:

1. Screen scraping: extract data from source code of website, with html parser and/or regular expressions
   - `rvest` package in R
2. Web APIs (application programming interfaces): a set of structured http requests that return JSON or XML data

   - `httr` package to construct API requests

Scraping the web: two approaches

Two different approaches:

1. Screen scraping: extract data from source code of website, with html parser and/or regular expressions
   - `rvest` package in R
2. Web APIs (application programming interfaces): a set of structured http requests that return JSON or XML data

   - `httr` package to construct API requests
   - Packages specific to each API: weatherData, WDI, Rfacebook... Check CRAN Task View on Web Technologies and Services for more examples

The rules of the game

1. Respect the hosting site's wishes:

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download

The rules of the game

1. Respect the hosting site's wishes:
    - First, check if an API exists or if data are available for download
    - Some websites *disallow* scrapers on their `robots.txt` files

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - Wait one or two seconds after each hit

The rules of the game

1. Respect the hosting site's wishes:
   - ▶ First, check if an API exists or if data are available for download
   - ▶ Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - ▶ Wait one or two seconds after each hit
   - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - Wait one or two seconds after each hit
   - Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

3. When using APIs, read documentation

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - Wait one or two seconds after each hit
   - Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

3. When using APIs, read documentation
   - Is there a batch download option?

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - Wait one or two seconds after each hit
   - Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

3. When using APIs, read documentation
   - Is there a batch download option?
   - Are there any rate limits?

The rules of the game

1. Respect the hosting site's wishes:
   - First, check if an API exists or if data are available for download
   - Some websites *disallow* scrapers on their `robots.txt` files

2. Limit your bandwidth use:
   - Wait one or two seconds after each hit
   - Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

3. When using APIs, read documentation
   - Is there a batch download option?
   - Are there any rate limits?
   - Can you share the data?

The art of web scraping

Workflow:

1. Learn about structure of website

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code
3. Generalize: functions, loops, debugging

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code
3. Generalize: functions, loops, debugging
4. Data cleaning

Three main scenarios
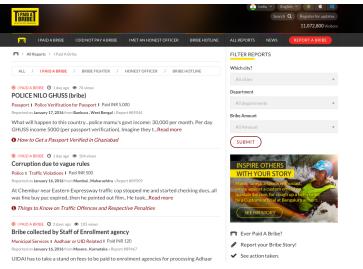
1. Data in table format

Article Talk

Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Main page

International court

From Wikipedia, the free encyclopedia

List of international courts [edit]

Name	Scope	Years active	Subject matter
International Court of Justice	Global	1945–present	General disputes
International Criminal Court	Global	2002–present	Criminal prosecutions
Permanent Court of International Justice	Global	1922–1946	General disputes
Appellate Body	Global	1995–present	Trade disputes within the WTO
International Tribunal for the Law of the Sea	Global	1994–present	Maritime disputes
African Court of Justice	Africa	2009–present	Interpretation of AU treaties
African Court on Human and Peoples' Rights	Africa	2006–present	Human rights
COMESA Court of Justice	Africa	1998–present	Trade disputes within COMESA
ECOWAS Community Court of Justice	Africa	1996–present	Interpretation of ECOWAS treaties
East African Court of Justice	Africa	2001–present	Interpretation of EAC treaties
SADC Tribunal	Africa	2005–2012	Interpretation of SADC treaties

Three main scenarios

2. Data in unstructured format

www.ipaidabribe.com/reports/paid

Three main scenarios

3. Data hidden behind web forms

Candidates on 2015 Venezuelan parliamentary election

Three main scenarios

1. Data in table format

Three main scenarios

1. Data in table format
   - Automatic extraction with `rvest`

Three main scenarios

1. Data in table format
   - Automatic extraction with `rvest`

2. Data in unstructured format

Three main scenarios

1. Data in table format
   - Automatic extraction with `rvest`

2. Data in unstructured format
   - Element identification with `selectorGadget`

Three main scenarios

1. Data in table format
   - Automatic extraction with `rvest`

2. Data in unstructured format
   - Element identification with `selectorGadget`
   - Automatic extraction with `rvest`

1. Data in table format
   - Automatic extraction with `rvest`

2. Data in unstructured format
   - Element identification with `selectorGadget`
   - Automatic extraction with `rvest`

3. Data hidden behind web forms

Three main scenarios

1. Data in table format
   - Automatic extraction with `rvest`

2. Data in unstructured format
   - Element identification with `selectorGadget`
   - Automatic extraction with `rvest`

3. Data hidden behind web forms
   - Automation of web browser behavior with `selenium`