Wrangling Court Data on a National Level

A presentation by

Mike Lissner

creator of

CourtListener.com

and

Juriscraper

The agenda

- Who am I?
- What is CourtListener?
- What is Juriscraper?
    - How does it work?
    - What does it do?
    - How can you contribute?
    - What's the future hold?

Me

- Mike Lissner
- Not:
  - A lawyer
  - A computer scientist
- Am:
  - Grad from UC Berkeley School of Information
  - Employee of a search company you *may* know
  - Open source/access enthusiast
- Have blog at http://michaeljaylissner.com

CourtListener Background

- Started in 2010
- Aggregates data and provides alerts
- Powerful search engine
- Data dumps
- Citation linking (see Rowyn's presentation!)
- Free. Free. Free.
- Demo

Juriscraper

- Our main topic du jour.
- A newer project used live on CourtListener
- A simple open source scraper that anybody can use

Juriscraper's Features

- Extensibility
- Solid, modern code
- Character detection and normalization
- Simple installation
- Harmonization
- Sophisticated title casing
- Sanity checking and hard failures

Extensibility

- Supports:
  - Varied geographies (countries, states, federal)
  - Languages
  - Media types (video, oral arguments, text)
- Currently has scrapers for:
  - Federal Appeals courts
  - Some states
  - Some special jurisdictions
  - Some back scrapers

Modern Code

- Requires: DRY, OO, PEP8
- Uses:
  - Python 2.7
  - lxml and XPath
  - Requests
  - chardet

Character Encodings

- Detects the declaration in XML or HTML pages
- If that's missing, then sniffs the encoding based on the binary data.
- Normalizes everything to UTF-8

Harmonization

- Words like, “et al, appellant, executor”, etc. all get removed.
- All forms of “USA” get normalized (U.S.A., U.S., United States, US, etc.)
- All forms of “vs” get normalized.
- Text gets titlecased if needed (much harder than it seems!)
- Junk punctuation gets removed/replaced
- Dates get converted to Python objects and results are guaranteed in reverse chronological order.

Sanity Checking and Hard Failures

- Court websites change frequently
- If our meta data is bad, we should fail completely and loudly

Integrating Juriscraper
aka
"All about the Caller"

- You have to build a "caller"
- You'll want:
  - Duplicate detection
  - Minimal impact on court websites
  - Mimetype detection
  - OCR
  - PDF "Decryption"

Duplicate Detection

- Test if the site has changed using a hash
- If so, extract the meta data from the page using Juriscraper.
- Iterate over the items, download their text or binary.
    - If a hash of the text or binary is new, save the item and proceed to the next
    - Else, dup_count++
- If proceeding, check the date of the next item.
    - If prior to the dup we found, terminate.
    - Else check a hash on the next item.
- If five dup_count == 5, terminate.

Impact Minimization

- Methods:
  - Reasonable duplicate detection algorithms
  - User-agent set to “juriscraper”
  - Free sharing of data via our API

Mimetypes, OCR and PDFs

- Mimetypes can be detected via “magic numbers”
- Text can then be extracted.
- If no text, use OCR.
- If text is garbled, try “decrypting” it

This would be awful, but…

We built a sample caller.

Two, actually.

Getting involved

- No more siloed scrapers!
- All code is open source (BSD license)
- Installation is simple (five minutes using pip)
- We built some custom tools to make development easier.
- Looking for:
  - More users
  - More developers

Why this is important

- Scaling is vital.
- More callers means:
  - More jurisdictions
  - Faster response times
  - Improved code
  - A unified court scraper (user-agent)

Juriscraper's Future

- Better alerts for downed scrapers
- Court-level rate throttling
- HTML tidying
- API Refactoring
- **More courts**!
- More backscrapers
- More unit tests

Juriscraper Demo/walkthrough

Thank you.

- http://courtlistener.com/
- https://bitbucket.org/mlissner/search-and-awareness-platform-courtlistener/
- https://bitbucket.org/mlissner/juriscraper/
- http://michaeljaylissner.com/