ILLUSTRATION BY THE PROJECT TWINS

GETTING GOOD DATA IN A SCRAPE

Public websites offer vast troves of data, but researchers must develop some fundamental software skills to make use of them. **By Michael Eisenstein**

When Ensheng Dong co-created the Johns Hopkins University COVID-19 Dashboard in January 2020, it was a labour of love. Dong, a systems engineer at the university in Baltimore, Maryland, had friends and family in China, including some in Wuhan, the site of the initial outbreak. "I really wanted to see what was going on in their area," he says. So Dong began collecting public-health data from the cities known to be affected.

At first, the work was manual. But as the outbreak became a pandemic, and the COVID-19 Dashboard became the go-to source for governments and scientists seeking information on the spread of the disease, Dong and his colleagues struggled to keep up. In the United States alone, the team was tracking medical reports from more than 3,000 counties, he says. "We were updating at least three to four times a day," he recalls, and there was no way the team could keep up that relentless pace manually. Fortunately, he and his graduate adviser, systems engineer Lauren Gardner, found a more scalable solution: web scraping.

Scraping algorithms pluck out relevant information from websites and report it in a spreadsheet or other user-friendly format. Dong and his colleagues developed a system that could capture COVID-19 data from around the world and update the numbers without human intervention. "For the first time in human history, we can track what's going on with a global pandemic in real time," he says.

Similar tools are harvesting data across a range of disciplines. Alex Luscombe, a criminologist at the University of Toronto in Canada, uses scraping to monitor Canadian law-enforcement practices; Phill Cassey, a conservation biologist at the University of Adelaide, Australia, tracks the global wildlife trade on Internet forums; and Georgia Richards, an epidemiologist at the University of Oxford, UK, scans coroners' reports for preventable causes of death. The technical skill required isn't trivial, but neither is it overwhelming — and the benefits can be immense, enabling researchers to collect large quantities of data rapidly without the errors inherent to manual transcription. "There's so many resources and so much information available online," Richards says. "It's just sitting there waiting for someone to come and make use of it."

Getting the goods

Modern web browsers are sufficiently polished that it's easy to forget their underlying

complexity. Websites blend code written in languages such as HTML and JavaScript to define where various text and visual elements will appear on the page, including both 'static' (fixed) content and 'dynamic' content that changes in response to user action.

Some scientific databases, such as PubMed, and social networks, such as Twitter, provide application programming interfaces (APIs) that offer controlled access to these data. But for other sites, what you see is what you get, and the only way to turn website data into something you can work with is by laboriously copying the visible text, images and embedded files. Even if an API exists, websites might limit which data can be obtained and how often.

Scrapers offer an efficient alternative. After being 'trained' to focus on particular elements on the page, these programs can collect data manually or automatically, and even on a schedule. Commercial tools and services often include user-friendly interfaces that simplify the selection of web-page elements to target. Some, such as the Web Scraper or Data Miner web browser extensions, enable free manual or automated scraping from small numbers of pages. But scaling up can get pricey: services such as Mozenda and ScrapeSimple charge a minimum of US$250 per month for scraping-based projects. These tools might also lack the flexibility needed to tackle diverse websites.

As a result, many academics prefer open-source alternatives. The Beautiful Soup package, which extracts information from HTML and XML files, and Selenium, which can also handle dynamic JavaScript content, are compatible with the Python programming language; rvest and RSelenium provide analogous functionality for R, another language. But these software libraries typically provide only the building blocks; researchers must customize their code for each website. "We worked with some of the pre-existing tools, and then we modified them," says Cassey of the scrapers he developed. "They've become increasingly bespoke through time."

Cracking the code

Simple web-scraping projects require relatively modest coding skills. Richards says her team resolves most problems "by Googling how to fix an error". But a good understanding of web design and coding fundamentals confers a valuable edge, she adds.

"I mostly use developer mode now," says Luscombe, referring to the browser setting that allows users to peel away a website's familiar façade to get at the raw HTML and other programming code below. But there are tools that can help, including the SelectorGadget browser extension, which provides a user-friendly interface to identify the 'tags' associated with specific website elements.

The complexity of a scraping project is largely determined by the site being targeted. Forums typically have fairly standard layouts, and a scraper that works on one can be readily tweaked for another. But other sites are more problematic. Cassey and his colleagues monitor sales of plants and animals that are either illegal or potentially harmful from an ecological perspective, and forums hosting such transactions can appear and disappear without warning, or switch their design. "They tend to be much more changeable to try to restrict the ease with which off-the-shelf web scrapers can just come through and gather information," says Cassey. Other websites might contain encrypted HTML elements or complex dynamic features that are difficult to decipher. Even sloppy web design can sabotage a scraping project — a problem that Luscombe often grapples with when scraping government-run websites.

"It becomes more of a data-processing problem than a problem of obtaining data."

The desired data might not be available as HTML-encoded text. Chaowei Yang, a geospatial researcher at George Mason University in Fairfax, Virginia, oversaw the development of the COVID-Scraper tool, which pulls pandemic case and mortality data from around the world. He notes that in some jurisdictions, these data were locked in PDF documents and JPEG image files, which cannot be mined with conventional scraping tools. "We had to find the tools that can read the data sets, and also find local volunteers to help us," says Yang.

Due diligence for data

Once you work out how to scrape your target site, you should give thought to how to do so ethically.

Websites typically specify terms of service that lay out rules for data collection and reuse. These are often permissive, but not always: Luscombe thinks that some sites weaponize terms to prevent good-faith research. "I work against tons of powerful criminal-justice agencies that really have no interest in me having data about the race of the people that they're arresting," he says.

Many websites also provide 'robots.txt' files, which specify acceptable operating conditions for scrapers. These are designed in part to prevent automated queries overwhelming servers, but generally leave wiggle room for routine data collection. Respecting these rules is considered best practice, even if it protracts the scraping process, for instance by building in delays between each page request. "We don't extract things at a rate faster than a user would," says Cassey. Researchers can also minimize server traffic by scheduling scraping jobs during off-peak hours, such as the middle of the night.

If private and personally identifiable data are being harvested, extra precautions might be required. Researchers led by Cedric Bousquet at the University Hospital of Saint-Étienne in France developed a tool called Vigi4Med, which scrapes medical forums to identify drug-associated adverse events that might have escaped notice during clinical testing. "We anonymized the user IDs, and it was separated from the other data," says Bissan Audeh, who helped to develop the tool as a postdoctoral researcher in Bousquet's lab. "The team that worked on data annotation didn't have any access to those user names." But context clues from online posts still potentially allow the re-identification of anonymized users, she says. "No anonymization is perfect."

Order from chaos

Scraping projects don't end when the harvesting is done. "All of a sudden, you're dealing with enormous amounts of unstructured data," says Cassey. "It becomes more of a data-processing problem than a problem of obtaining data."

The Johns Hopkins COVID Dashboard, for instance, requires careful fact-checking to ensure accuracy. The team ended up developing an anomaly-detection system that flags improbable shifts in numbers. "Say a small county that used to report 100 cases every day reports maybe 10,000 cases," says Dong. "It might happen, but it's very unlikely." Such cases trigger closer inspection of the underlying data — a task that depends on a small army of multilingual volunteers who can decipher each nation's COVID-19 reports. Even something as simple as a typo or change in how dates are formatted can gum up a data-analysis pipeline.

For Cassey's wildlife-tracking application, determining which species are actually being sold — and whether those transactions are legal — keeps the team on its toes. If sellers know they're breaking the law, they will often obfuscate transactions with deliberately misleading or street names for plants and animals, much like online drug dealers do. For one particular parrot species, for instance, the team has found 28 'trade names', he says. "A lot of fuzzy data matching and natural-language processing tools are required."

Still, Richards says would-be scrapers shouldn't be afraid to explore. Start by repurposing an existing web scraper. Richards' team adapted its software for analysing coroners' reports from a colleague's tool for clinical-trials data. "There's so many platforms out there and there's so many online resources," she says. "Just because you don't have a colleague that has web-scraped before, don't let that prevent you from giving it a go."

Michael Eisenstein is a science writer in Philadelphia, Pennsylvania.