Digging Deeper Reaching Further

Libraries Empowering Users to Mine the HathiTrust Digital Library Resources

Module 2.2 Gathering Textual Data: Bulk retrieval Lesson Plan

Further reading: go.illinois.edu/ddrf-resources

This lesson covers methods for gathering textual data from the web in bulk, including using APIs, file transfers, and web scraping, and also introduces the command line interface.

Estimated time

45-60 minutes

Audience

Librarians with some exposure to text analysis who may be supporting text analysis research at their institutions.

Prerequisites for participants

- Have some idea of text analysis concepts
- Have been introduced to the HTRC, or have completed Module 1
- Have been introduced to the concept of text as data in digital scholarship and are familiar with the options available to researchers for accessing textual data, or have competed Module 2.1

Learning objectives

At the end of the module, the participants will be able to:

- Execute basic commands from the command line interface in order to gain confidence with computationally-intensive research.
- Understand why automated access is valuable for building textual datasets in order to facilitate researcher needs around digital scholarship.

Skills

- Command line
- Execute a web scraping command

Getting ready

Workshop participants will need:

- Access to a computer, the Internet, and a web browser
- Access to PythonAnywhere and an account

Session outline

- Introduction to bulk retrieval and bulk HTRC data
- Introduction to methods of automating bulk retrieval
    - Web scraping
    - APIs
    - Transferring files
- **Activity:** Explore the basic HathiTrust Bibliographic API
- Introduction to the command line
- **Activity:** Run basic Bash commands
- **Activity:** Scrape a webpage
- Creativity Boom case study: How Sam did bulk HTRC data retrieval
- Discussion: Does your library provide access to digitized materials in a way that is conducive to text analysis?

Key concepts

- **Command line**: A text-based interface that takes in commands and passes them to the computer's operating system. Commands can be used to accomplish (and script) a wide range of tasks. The interface is often called a **shell**, such as the **Bash shell**.
- **API (Application Programming Interface)**: A set of clearly-defined communication methods (may include commands, functions, protocols, objects, etc.) that can be used to interact with an external system. They are basically instructions (written in code) for accessing systems or collections.
- **Script**: A file containing a set of programing statements that can be run using the command line.

- **Web scraping:** The process of extracting data from webpages.

Key tools

- **File Transfer Protocol (FTP):** A protocol that computers on a network use to transfer files to and from each other. A protocol is a set of rules that networked computers use to talk to one another, like a language.
- **Secure/SSH File Transfer Protocol (SFTP):** Works in a way similar to FTP, but is a separate protocol that encrypts the connection to enable a secure file transfer.
- **rsync:** A fast file-copying tool widely used for backups. It's well-known for its efficiency, because it reduces the amount of data sent over the network by sending only the differences between the files at the source location and the files at the destination location.
- **PythonAnywhere:** A browser-based programming environment that's also a code editor and file hosting service. It comes with a built-in Bash shell and does not interact with your local file system.
- **wget:** A command line tool for retrieving files from a server. It can scrape the contents of a website, with options that can be modified to tailor more specifically to how you want the contents to be retrieved.
- **Beautiful Soup:** A Python-based web scraping tool that pulls data out of HTML and XML files. It has several options for specifying what you want to scrape (within the HTML) and is good for getting clean, well-structured text.

Key points

Introduction to bulk retrieval and bulk HTRC data	• Gathering large amounts of textual data is a time-consuming process – it's necessary to automate retrieval when possible. • Some HT and HTRC datasets can be retrieved using APIs and rsync.
Introduction to methods of automating bulk retrieval	• Some methods for automating retrieval are: web scraping using tools or via running commands/scripts; using APIs; transferring files with FTP, SFTP, or rsync.

Activity: Use an API	• Retrieve metadata using the HathiTrust's Bibliograpic API. • *Goal:* Demystify data APIs to show how they facilitate data transfer.
Introduction to the command line	• The command line is a text-based interface that takes in commands and passes them on to the computer's operating system to accomplish tasks. • You can use a web-based tool called PythonAnywhere with a built-in Bash shell to run commands and scripts.
Activity: Run basic Bash commands	• Use video to introduce some basic Bash commands, such as "pwd" and "cd", and guide participants in practicing them in PythonAnywhere. Participants will also unzip and move the activity files that will be used in later activities. • *Goal:* Gain hands-on experience with the command line in preparation for the following activity.
Activity: Run wget to scrape a webpage	• Guide participants in running a command on PythonAnywhere that scrapes the text from a webpage version of George Washington's *Fourth State of the Union Address*. • Review the scraped text, summarize the process, and discuss next steps. • On their own, participants revise the command to scrape George Washington's *Second State of the Union Address.* • *Goal:* Build confidence on the command line and show how automated data retrieval makes it easier to grab data than manual copying.
Creativity Boom case study	• Sam used rsync to bulk retrieve HTRC Extracted Features files.
Discussion	• Question: Does your library provide access to digitized materials in a way that is conducive to text analysis?

	• *Goal:* Prompt librarians to consider how library collections are data sources for text analysis.

Additional Tips for Instructors

- **Recommend participants NOT to use Internet Explorer for the web-based activities and choose an alternative browser such as Chrome or Firefox.** Participants using IE may encounter some issues with some of the activities.
- When demonstrating the commands in PythonAnywhere, instructors may use “Ctrl” and “+” (“Command” and “+” on Macs) to enlarge the content on the screen. It can be very difficult to see the command line from the back of the room! Use “Ctrl” and “-” (“Command” and “-” on Macs) to zoom back out when you need to demonstrate other things in regular size.
- It could be helpful to have at least two instructors teaching this module, with one demonstrating commands and running scripts in the front, and the another moving around the room to help participants troubleshoot any issues.