Analyzing Web Archives with the Archives Unleashed Project

Samantha Fritz, MLIS
Project Manager
Archives Unleashed
sam.fritz@archivesunleashed.org

Ian Milligan, PhD
Associate Professor of History
University of Waterloo
i2millig@uwaterloo.ca
Overview

- Web Archiving Context
- Archives Unleashed Project
- Archives Unleashed Toolkit
  - Setup
  - Hands-On Activities
- External Tools: Voyant & Gephi
- Wrap Up
Web Archiving Context
Web Archiving

Web Archiving is the deliberate process of preserving born-digital content on the World Wide Web.
Web Archiving

The Web has shaped how we connect with one another and interact with information.

Photo by Robynne Hu on Unsplash
4.66 BILLION internet users

In 2020
1.7MB of data created / sec / person
Web Archiving

The web continues to grow at an exponential rate
Web Archiving

The web continues to grow at an exponential rate

BUT

The web is also disappearing

Photo by Erik Mclean on Unsplash
Web Archiving

Allows us to preserve vulnerable cultural information in the form of born-digital artifacts

6,104,790 #WomensMarch images

Full dataset is available here.

Created with Juxta.

Exploring #WomensMarch.

https://ruebot.net/visualizations/wm/
Web Archiving

1991  WWW made publicly available

1996  First large scale preservation projects initiated

2021  Petabytes of data for studying topics from the 1990s forward

Photo by Jason Leung on Unsplash
Barriers to Web Archives

- Abundance of data is a challenge and overwhelming
- Understanding of high-performance computing
- Familiarity with command line
- Inadequacies of time, resources, support
How do we lower this barrier to access and use of web archives?
Established in 2017 to create accessible and user-friendly tools to work with web archives.
Archives Unleashed Project

Looking for a way to explore web archives through a user-friendly suite of tools?

AU Toolkit  AU Cloud  Warclight  Notebooks

https://archivesunleashed.org
Archives Unleashed Project

Platform for analyzing web archives built on Hadoop and Apache Spark.

Resources:

Toolkit User Documentation
https://aut.docs.archivesunleashed.org
Archives Unleashed Toolkit Workshop
Activity Plan

We will run through a few activities and scripts to get you working with the Archives Unleashed Toolkit, as well as demonstrate how information extracted from the Toolkit can also be used with external tools for further analysis.

Overview

1. Setup Docker
2. Launch Archives Unleashed Toolkit
3. Run Scripts & Dig into WARCs
The example data used in this workshop is drawn from the Canadian Political Parties & Political Interest Groups Archive-It Collection.

This collection was curated by the University of Toronto.

https://archive-it.org/collections/227
For each of these slides, we will present the “concepts” and then provide a short video showing us putting them into action.
Setup Docker
“Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Citation: “What is Docker” https://opensource.com/resources/what-docker
Setup Docker

“Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Citation: “What is Docker” https://opensource.com/resources/what-docker
Setup Docker

“Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Citation: “What is Docker” https://opensource.com/resources/what-docker

Step 3: Run through Docker install

1. Double click .dmg folder to launch install

2. Drag and drop into Applications folder

3. Open Docker

NOTE: Docker may require access depending on system requirements
Setup Docker

“Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Citation: “What is Docker” https://opensource.com/resources/what-docker
Setup Docker

“Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Citation: “What is Docker” https://opensource.com/resources/what-docker

Step 3: Check Docker is running

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker version</td>
<td>to check that you have the latest release installed</td>
</tr>
<tr>
<td>docker run hello-world</td>
<td>to verify that Docker is pulling images and running as expected</td>
</tr>
</tbody>
</table>
Launch Archives Unleashed Toolkit (AUT)
Launching AUT

- Make a Directory
- Launch Spark Shell
- Tips on Using the Shell

Create a directory (folder) on your desktop and call it data.

Note: You can do this in terminal using the commands below, or right click on the desktop and create new folder.

```
cd desktop
mkdir data
```

Resources:

Toolkit User Documentation
https://aut.docs.archivesunleashed.org

Note the path: e.g. /Users/ianmilligan1/desktop/data
Launching AUT

- Make a Directory
- Launch Spark Shell
- Tips on Using the Shell

This will launch the Apache Spark Shell and makes the connection between the directory called "data" on the desktop with a directory in the Docker virtual machine.

Resources:

Toolkit User Documentation
https://aut.docs.archivesunleashed.org

Script

docker run --rm -it -v "/path/to/your/data:/data"
archivesunleashed/docker-aut

Be sure to change the path!
Launching AUT

- Make a Directory
- Launch Spark Shell
- Tips on Using the Shell

Reminder: use a text editor for copy/paste/edit of scripts, to avoid any text formatting issues (e.g. curly quotes).

Resources:

Toolkit User Documentation
https://aut.docs.archivesunleashed.org

Welcome to Spark Shell!

Before we start using scripts, a few things to note about using Spark Shell.

1) To copy and use scripts

:paste

2) To exit from paste mode

ctrl + D

3) To exit AUT completely

ctrl + D
Run Scripts & Dig into WARC}s
**Archives Unleashed Project**

- **Collections Analytics**
  - List of URLs
- Plain Text Extraction
  - All plain text
  - Plain text by domain
- Analysis of Site Link Structure
  - Exporting to Gephi Directly
- Image Analysis
  - Most frequent image URLs in a collection
- Example using DF (Dataframes)
  - Top Domains
  - Image Analysis

```scala
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

val r = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.gz", sc)
  .keepValidPages()
  .map(r => ExtractDomain(r.getUrl))
  .countItems()
  .take(10)
```

Hello World!
This Script:

- Imports the AUT libraries;
- Tells the program where it can find the data;
- Tells it only to keep the "valid" pages, in this case HTML data;
- Tells it to ExtractDomain, or find the base domain of each URL;
- Count them - how many times a URL appears in a collection,
- Display the top ten!

This script is used to:

- Simple & lets us know that AUT is working;
- It also helps us to understand what we can expect to find in the web archives!

```scala
import io.archivesunleashed._
import io.archivesunleashed.matchbox._
val r = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.*.gz", sc)
  .keepValidPages()
  .map(r => ExtractDomain(r.getUrl))
  .countItems()
  .take(10)
```
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

val r = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.{gz}", sc)
  .keepValidPages()
  .map(r => ExtractDomain(r.getUrl))
  .countItems()
  .take(10)
Archives Unleashed Project

- Collections Analytics
  - List of URLs (some or all)
- Plain Text Extraction
  - All plain text
  - Plain text by domain
- Analysis of Site Link Structure
  - Exporting to Gephi Directly
- Image Analysis
  - Most frequent image URLs in a collection
- Example using DF (Dataframes)
  - Top Domains
  - Image Analysis

Script

```scala
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

RecordLoader.loadArchives("/aut-resources/Sample-Data/*\.gz", sc)
  .keepValidPages()
  .map(r => (r.getCrawlDate, r.getDomain, r.getUrl, RemoveHTML(r.getContentString)))
  .saveAsTextFile("/data/plain-text")
```

Output - pulls all the text and saves as a text file in our data folder.
Let's see it in action!

WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/spark/jars/spark-unsafe_2.12-3.0.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/03/09 21:12:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://c56f58f8c32:4040
Spark context available as 'sc' (master = local[*], app id = local-1615324335015).
Spark session available as 'spark'.
Welcome to

```
Spark version 3.0.1
```

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.8.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 
scala> 

Archives Unleashed Project

- Collections Analytics
  - List of URLs (some or all)
- Plain Text Extraction
  - All plain text
  - **Plain text by domain**
- Analysis of Site Link Structure
  - Exporting to Gephi Directly
- Image Analysis
  - Most frequent image URLs in a collection
- Example using DF (Dataframes)
  - Top Domains
  - Image Analysis

---

**Script**

```python
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

RecordLoader.loadArchives("/aut-resources/Sample-Data/*.gz", sc)
  .keepValidPages()
  .keepDomains(Set("www.liberal.ca"))
  .map(r => (r.getCrawlDate, r.getDomain, r.getUrl, RemoveHTML(RemoveHTTPHeader(r.getContentString))))
    .saveAsTextFile("/data/liberal-plain-text/")
```

**Output** - pulls all the text from a specific base domain and save as a text file in our data folder.
Archives Unleashed Project

We have several scripts that allow you to filter within the plain text.

- Plain Text Without HTTP Headers
- Plain Text by Domain
- Plain Text by URL Pattern
- Plain Text Minus Boilerplate
- Plain Text Filtered by Date
- Plain Text Filtered by Language
- Plain Text Filtered by Keyword

You may also choose to include (keep) or exclude (discard) specific filters.

** URLs → .keepUrlPatterns  

```
URLs: val urls = Set("archive.org","uwaterloo.ca","yorku.ca")
------------------------------------------
Example:
.keepUrlPatterns(Set("www.davidsuzuki.org"))
```

** Date Filter (full or partial) → .keepDate  

```
Dates: val dates = List("2008","200908","20070502")
------------------------------------------
Example:
.keepDate(List("200804"),
ExtractDate.DateComponent.YYYYMM)
```

```
Date Filter (full or partial) → .keepDate
```

** Language → .keepLanguages  

```
Languages: val languages = Set("en")
------------------------------------------
Example:
.keepLanguages(Set("en"))
```

Uses the ISO 639.2 language codes
import io.archivesunleashed._
import io.archivesunleashed.udfs._
import io.archivesunleashed.app._

val graph = RecordLoader.loadArchives("aut-resources/Sample-Data/*.*",sc)
  .webgraph.groupBy($"crawl_date",
    removePrefixWWW(extractDomain($"src")),
    removePrefixWWW(extractDomain($"dest")))
  .count()
  .filter(!$"dest_domain"=="")
  .filter(!$"src_domain"=="")
  .filter($"count" > 5)
  .orderBy(desc("count"))
  .collect()

WriteGEXF(graph, ":data/links-for-gephi.gexf")
Archives Unleashed Project

- Collections Analytics
  - List of URLs (some or all)
- Plain Text Extraction
  - All plain text
  - Plain text by domain
- Analysis of Site Link Structure
  - Exporting to Gephi Directly
- Image Analysis
  - Most frequent image URLs in a collection
- Example using DF (Dataframes)
  - Top Domains
  - Image Analysis

Script

```scala
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

val links = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.*.gz", sc)
  .keepValidPages()
  .flatMap(r => ExtractImageLinks(r.getUrl, r.getContentString))
  .countItems()
  .take(20)
```

Output - provides list of most frequent image URL

```scala
```
Archives Unleashed Project

- Collections Analytics
  - List of URLs (some or all)
- Plain Text Extraction
  - All plain text
  - Plain text by domain
- Analysis of Site Link Structure
  - Exporting to Gephi Directly
- Image Analysis
  - Most frequent image URLs in a collection
- Example using DF (Dataframes)
  - Top Domains
  - Image Analysis

Image URL in WayBack Machine
http://www.liberal.ca/shared/images/logo_footer.png

Visit: http://web.archive.org

Enter in the URL to see the use history/temporal distribution
import io.archivesunleashed._
import io.archivesunleashed.udfs._

RecordLoader.loadArchives("/aut-resources/Sample-Data/*.*.gz", sc).webpages().
  .select(extractDomain($"url").as("domain"))
  .groupBy("domain").count().orderBy(desc("count"))
  .show(20, false)
import io.archivesunleashed._
import io.archivesunleashed.udfs._

val df = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.*", sc).images();

df.select("url", "filename", "extension", "mime_type_web_server", "mime_type_tika", "width", "height", "md5", "sha1", "bytes")
  .orderBy(desc("md5"))
  .show()
Archives Unleashed Toolkit

For more examples of scripts to use with the Toolkit, please visit the User Documentation.

https://aut.docs.archivesunleashed.org
Hands-on: External Tools
Voyant Tools is a free web-based text analysis platform. Voyant allows you to quickly and easily visualize your data and export the visualizations for further use.

Voyant Tools: voyant-tools.org
Voyant Tools

Voyant Tools is a free web-based text analysis platform. Voyant allows you to quickly and easily visualize your data and export the visualizations for further use.

Voyant Tools: voyant-tools.org
Earlier, we extracted all text from the captures of the liberal.ca website within our sample data and generated a .txt file with all of this content.

This .txt file can be uploaded to Voyant to perform some basic analysis.
Voyant’s Links tool allows you to visualize context relationships between frequently-used keywords.
Voyant’s Trends tool allows you to graph the frequency of a keyword throughout your text file.
Voyant’s Context tool allows you to quickly view on a keyword and several words to the right and left. Clicking on a keyword instance will pull up that section of the text in the Reader view.
Gephi

Open source visualization and exploration software.

Archives Unleashed Learning Guide: Network Graphing Archived Websites with Gephi

https://cloud.archivesunleashed.org/derivatives/gephi

Gephi can be downloaded and installed from gephi.org
Gephi

Open source visualization and exploration software.

Archives Unleashed Learning Guide: Network Graphing Archived Websites with Gephi
https://cloud.archivesunleashed.org/derivatives/gephi

Step 1: Open the .gexf file generated by the Archives Unleashed Toolkit.
Gephi

Open source visualization and exploration software.

Archives Unleashed Learning Guide: Network Graphing Archived Websites with Gephi
https://cloud.archivesunleashed.org/derivatives/gephi

Step 2: Use Statistics and Filters tools to organize data.
Wrap Up
Final Thoughts

Resources

AUT Documentation
https://aut.docs.archivesunleashed.org

Additional Learning Resources
https://cloud.archivesunleashed.org/derivatives

Sample Projects from Datathons
https://archivesunleashed.org/events/

Project Links
Website  https://archivesunleashed.org
 Github  https://github.com/archivesunleashed
 Slack  http://slack.archivesunleashed.org/
 Twitter  @unleasharchives
 YouTube  UC4Sg0Xi6UWhYK2VbmAzFhAw

- Web archives are an important data source for those studying topics post-1990;
- It’s critical to provide researchers and scholars methods and tools to access and use web archives;
- The Archives Unleashed Toolkit provides transparent and flexible options for exploring web archives!
Sources

Software Mentioned

Docker
WayBack Machine
Voyant Tools
Gephi

https://www.docker.com
https://archive.org/web/
https://voyant-tools.org
https://gephi.org

Example Dataset

Canadian Political Parties & Political Interest Groups Archive-It Collection. https://archive-it.org/collections/227