



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.



Photo by [Everyday basics](#) on Unsplash

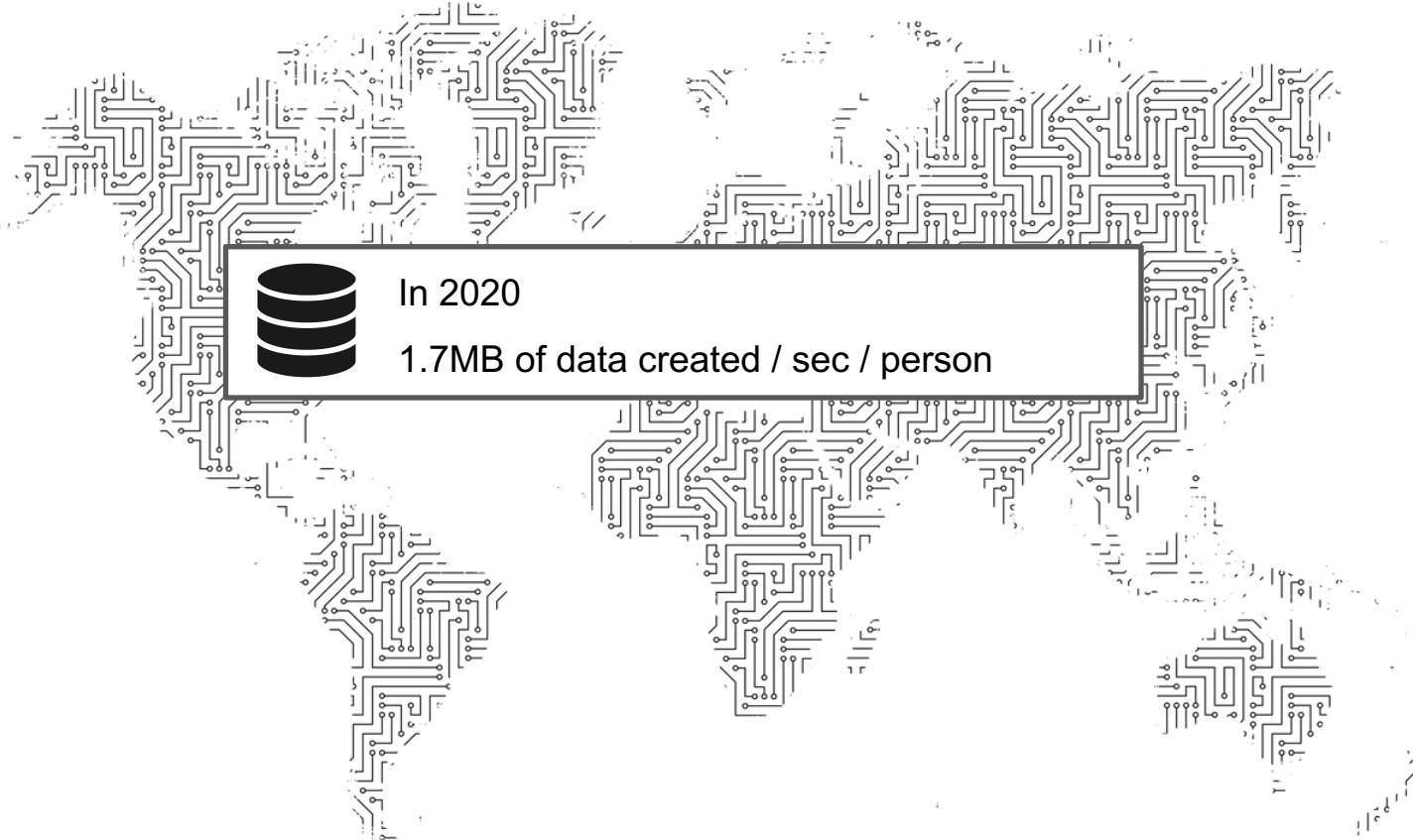
Web Archiving

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



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.



Generated 2017-11-06 22:35

Nick Ruest and Toke Eskildesen, Web Archives for Historical Research

<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

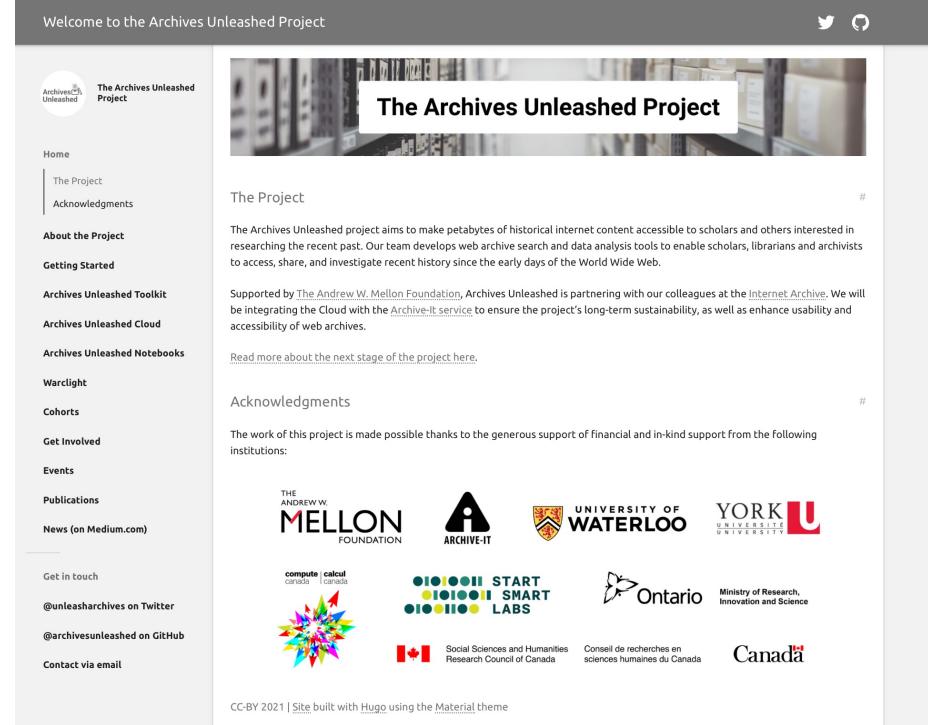


How do we lower this barrier to
access and **use** of web archives?

Archives Unleashed Project

Archives Unleashed Project

Established in 2017 to create
accessible and user-friendly
tools to work with web
archives.



The screenshot shows the homepage of the Archives Unleashed Project. At the top, there's a navigation bar with links for Home, The Project, Acknowledgments, About the Project, Getting Started, Archives Unleashed Toolkit, Archives Unleashed Cloud, Archives Unleashed Notebooks, Warclight, Cohorts, Get Involved, Events, Publications, and News (on Medium.com). Below the navigation is a main content area featuring a banner image of a library or archive room with the text "The Archives Unleashed Project". To the right of the banner is a section titled "The Project" which includes a brief description of the project's goal to make historical internet content accessible, supported by the Andrew W. Mellon Foundation and the Internet Archive. It also mentions the integration of the Archive-It service. Below this is a "Read more about the next stage of the project here." link. Further down is an "Acknowledgments" section listing financial support from the Andrew W. Mellon Foundation, University of Waterloo, York University, Compute Canada, START, SMART LABS, Ontario Ministry of Research, Innovation and Science, Social Sciences and Humanities Research Council of Canada, and Conseil de recherches en sciences humaines du Canada. At the bottom, there's a footer note about the site being built with Hugo using the Material theme.

Welcome to the Archives Unleashed Project

[The Archives Unleashed Project](#)

Home

The Project

Acknowledgments

About the Project

Getting Started

Archives Unleashed Toolkit

Archives Unleashed Cloud

Archives Unleashed Notebooks

Warclight

Cohorts

Get Involved

Events

Publications

News (on Medium.com)

Get in touch

@unleasharchives on Twitter

@archivesunleashed on GitHub

Contact via email

The Archives Unleashed Project

The Archives Unleashed project aims to make petabytes of historical internet content accessible to scholars and others interested in researching the recent past. Our team develops web archive search and data analysis tools to enable scholars, librarians and archivists to access, share, and investigate recent history since the early days of the World Wide Web.

Supported by The Andrew W. Mellon Foundation, Archives Unleashed is partnering with our colleagues at the Internet Archive. We will be integrating the project with the Archive-It service to ensure the project's long-term sustainability, as well as enhance usability and accessibility of web archives.

[Read more about the next stage of the project here.](#)

Acknowledgments

The work of this project is made possible thanks to the generous support of financial and in-kind support from the following institutions:

THE ANDREW W. MELLON FOUNDATION

UNIVERSITY OF WATERLOO

YORK UNIVERSITY

compute canada | calcul canada

START

SMART LABS

Ontario Ministry of Research, Innovation and Science

Social Sciences and Humanities Research Council of Canada

Conseil de recherches en sciences humaines du Canada

Canada

CC-BY 2021 | Site built with [Hugo](#) using the [Material](#) theme



Archives Unleashed Project

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



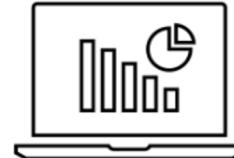
AU Toolkit



AU Cloud



Warlight



Notebooks

Archives Unleashed Project

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

```
Welcome to
   _/\_ 
  / \ \_ 
 /   \ \_ 
 \   / \_ 
  \ / \ \_ 
   \ \ \_ 
    \ \_ 
     \_ 
version 2.4.3

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_212)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 
```

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

```
Welcome to
   _/\_ _/\_ _/\_ _/\_ _/\_
  / \ \ / \ \ / \ \ / \ \ / \
 / \ \ / \ \ / \ \ / \ \ / \
version 2.4.3

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_212)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Sample Data Acknowledgement

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.

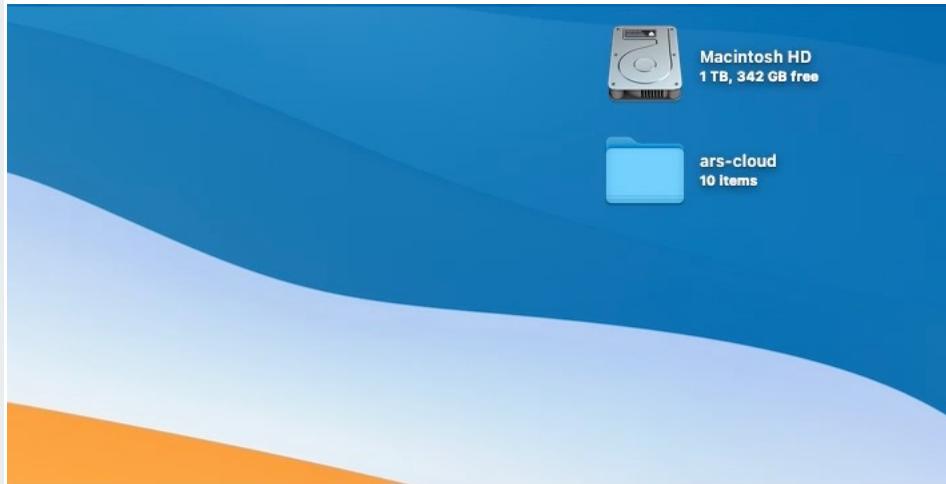
The screenshot shows the Archive-It website interface. At the top, there is a navigation bar with links for HOME, EXPLORE, LEARN MORE, and CONTACT US. On the right side of the header, there is a logo for the Internet Archive and text stating "The leading web archiving service for collecting and accessing cultural heritage on the web Built at the Internet Archive". Below the header, the URL "Explore >> University of Toronto >> Canadian Political Parties and Political Interest Groups" is displayed. The main content area features a banner for the "Canadian Political Parties and Political Interest Groups" collection, which was "Collected by: University of Toronto" and "Archived since: Oct, 2005". The banner also includes a description of the collection's purpose and rights information. Below the banner, there is a section titled "Narrow Your Results" with filters for "Group", "Sort By: Count | (A-Z)", and "Quarterly Crawl Brozzler (1)". There are also sections for "Subject" and "Creator" with similar filtering options. A search bar is present with the placeholder "Enter search terms here". The results page shows a list of items, with the first item being "Title: Cosmopolitan Party of Canada" with a URL of <http://agoracosmopolite.com/>. The results page indicates "Page 1 of 2 (125 Total Results)" and provides "Sort By" options: Title (A-Z), Title (Z-A), URL (A-Z), and URL (Z-A). The bottom of the page shows other items in the collection, such as "Title: Bloc Québécois" and "Title: Canada First Immigration Reform Committee".

<https://archive-it.org/collections/227>

How We'll Do It

Like so!

For each of these slides, we will present the “concepts” and then provide a short video showing us putting them into action.



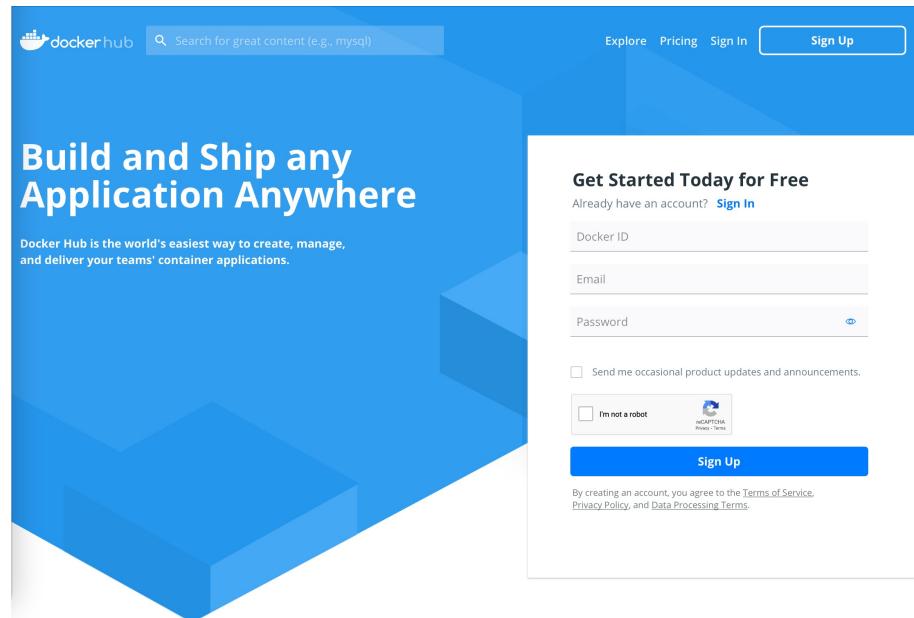
Setup Docker

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.”

Step 1: Sign Up for free Docker ID



<https://hub.docker.com>

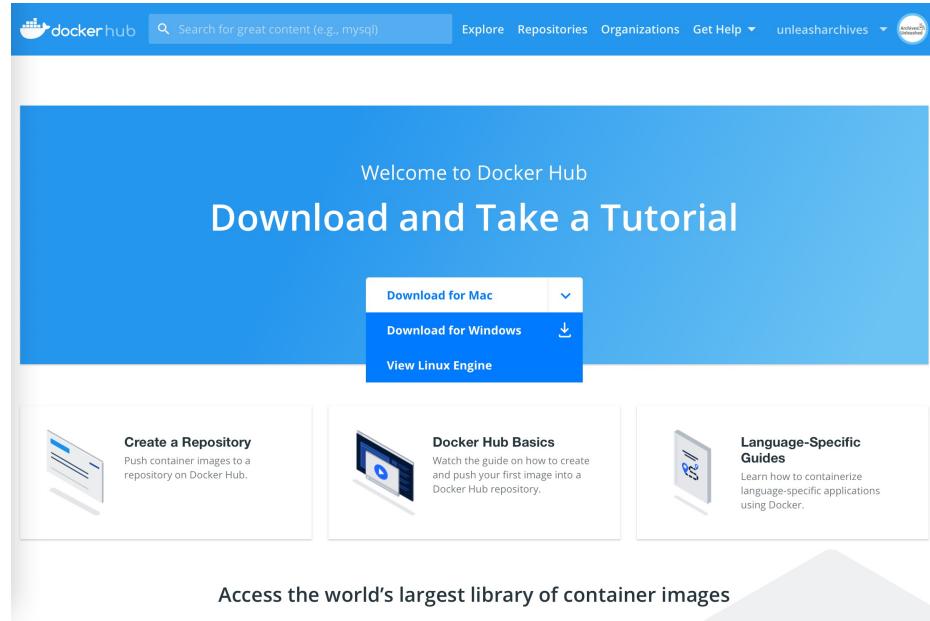
Setup Docker

“[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.”

Step 2: Login to Docker + Select OS System

<https://www.docker.com>



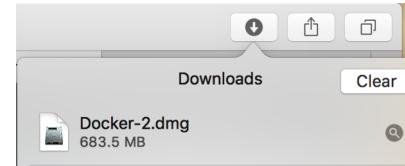
Setup Docker

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

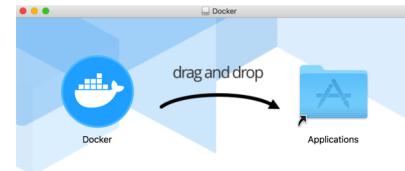
Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”

Step 3: Run through Docker install

1. Double click .dmg folder to launch install

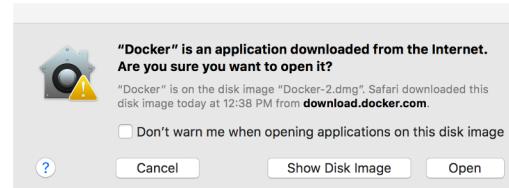


2. Drag and drop into Applications folder



3. Open Docker

NOTE: Docker may require access depending on system requirements



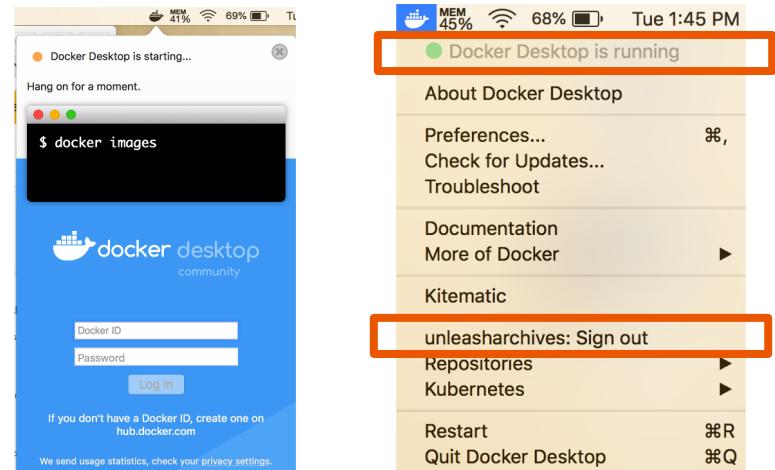
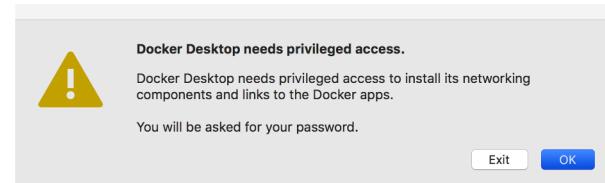
Setup Docker

“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.”

Step 3: Run through Docker install (cont.)

Open Docker - green dot indicates Docker is running



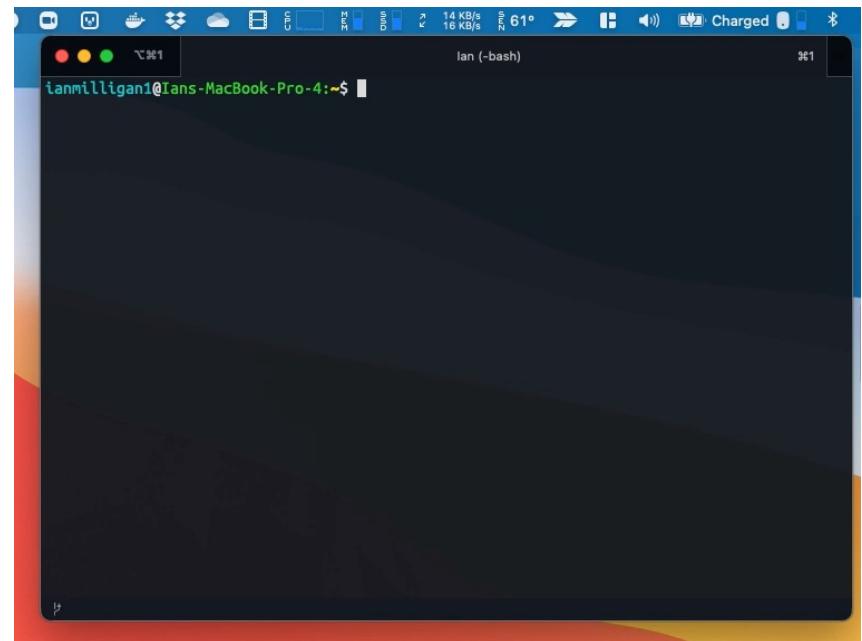
Setup Docker

“[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.”

Step 3: Check Docker is running

Command	Purpose
<code>docker version</code>	to check that you have the latest release installed
<code>docker run hello-world</code>	to verify that Docker is pulling images and running as expected



Launch Archives Unleashed Toolkit (AUT)

Launching AUT

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

Resources:

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

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
```



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:

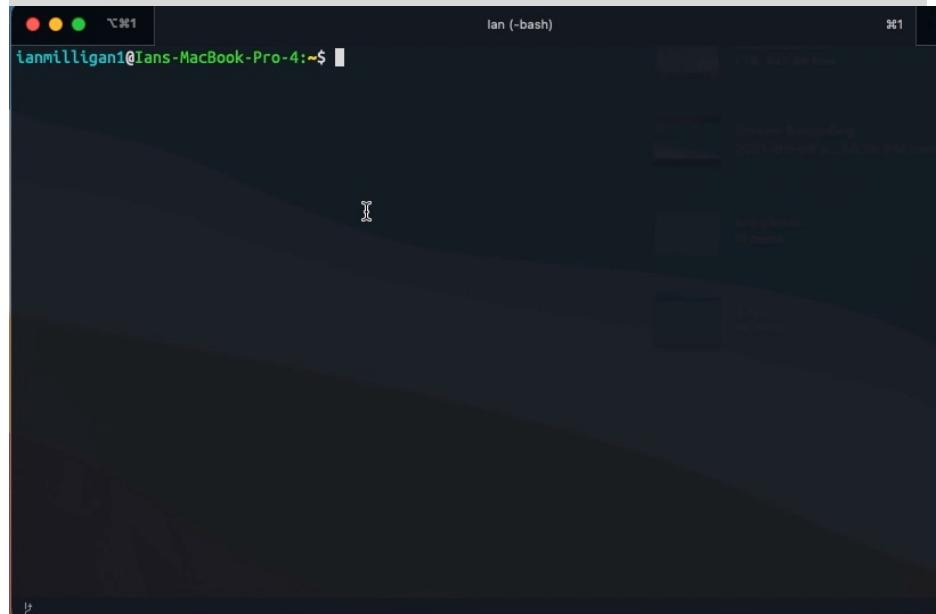
Toolkit User Documentation
<https://aut.docs.archivesunleashed.org>

Script

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

Be sure to change the path!

```
docker run --rm -it -v  
"/Users/ianmilligan1/desktop/data":/data"  
archivesunleashed/docker-aut
```



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 WARCs

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

Hello World!

```
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

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!

Hello World!

```
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)
```

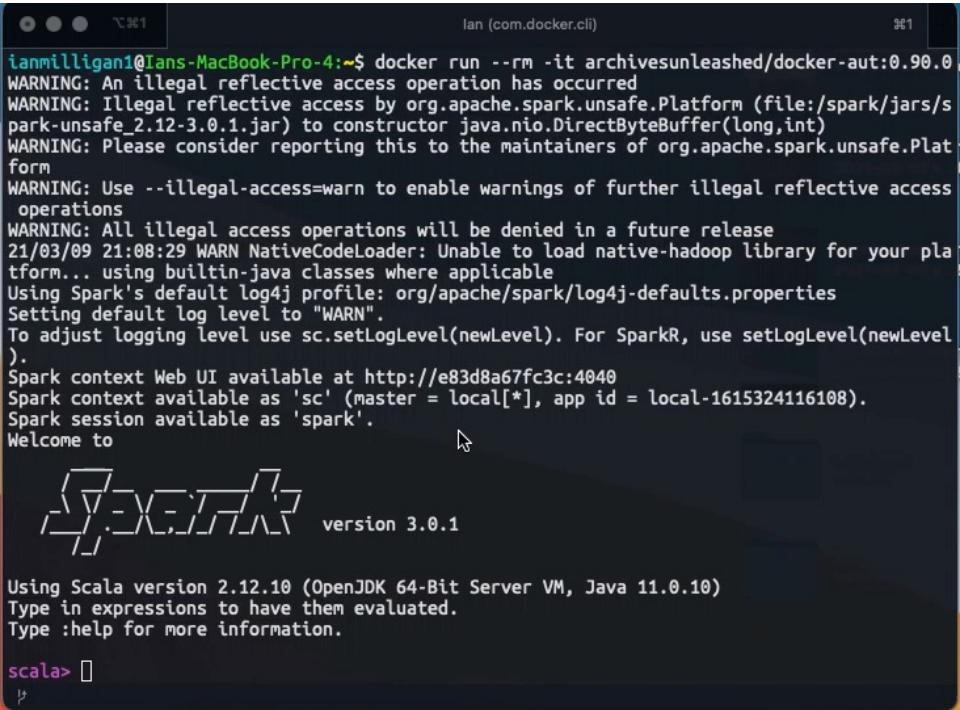
Archives Unleashed Project

Your turn to try!

:paste

```
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)
```

CTRL + D



A screenshot of a terminal window titled "ian (com.docker.cli)". The terminal shows Scala code being run against a Docker container. The code imports `io.archivesunleashed` and `io.archivesunleashed.matchbox` packages, defines a variable `r` using `RecordLoader.loadArchives` with a path to "Sample-Data/*.gz" and a `sc` (SparkContext) object, and then performs several operations: `keepValidPages`, `map`ing `r` to `ExtractDomain(r.getUrl)`, `countItems`, and `take(10)`. The output of the command is a series of warning messages from the Docker container, followed by the Scala REPL prompt "scala>".

```
ianmilligan1@Ians-MacBook-Pro-4:~$ docker run --rm -it archivesunleashed/docker-aut:0.90.0  
WARNING: An illegal reflective access operation has occurred  
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 21:08:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-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://e83d8a67fc3c:4040  
Spark context available as 'sc' (master = local[*], app id = local-1615324116108).  
Spark session available as 'spark'.  
Welcome to  
version 3.0.1  
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)  
Type in expressions to have them evaluated.  
Type :help for more information.  
scala>   
^C
```

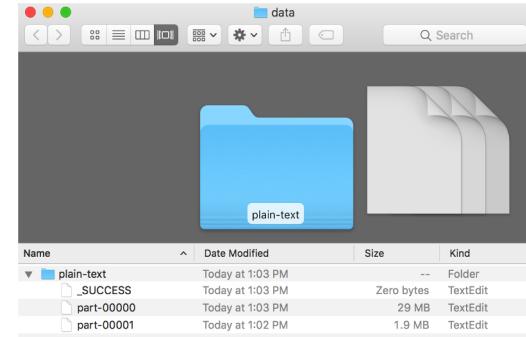
Archives Unleashed Project

- 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

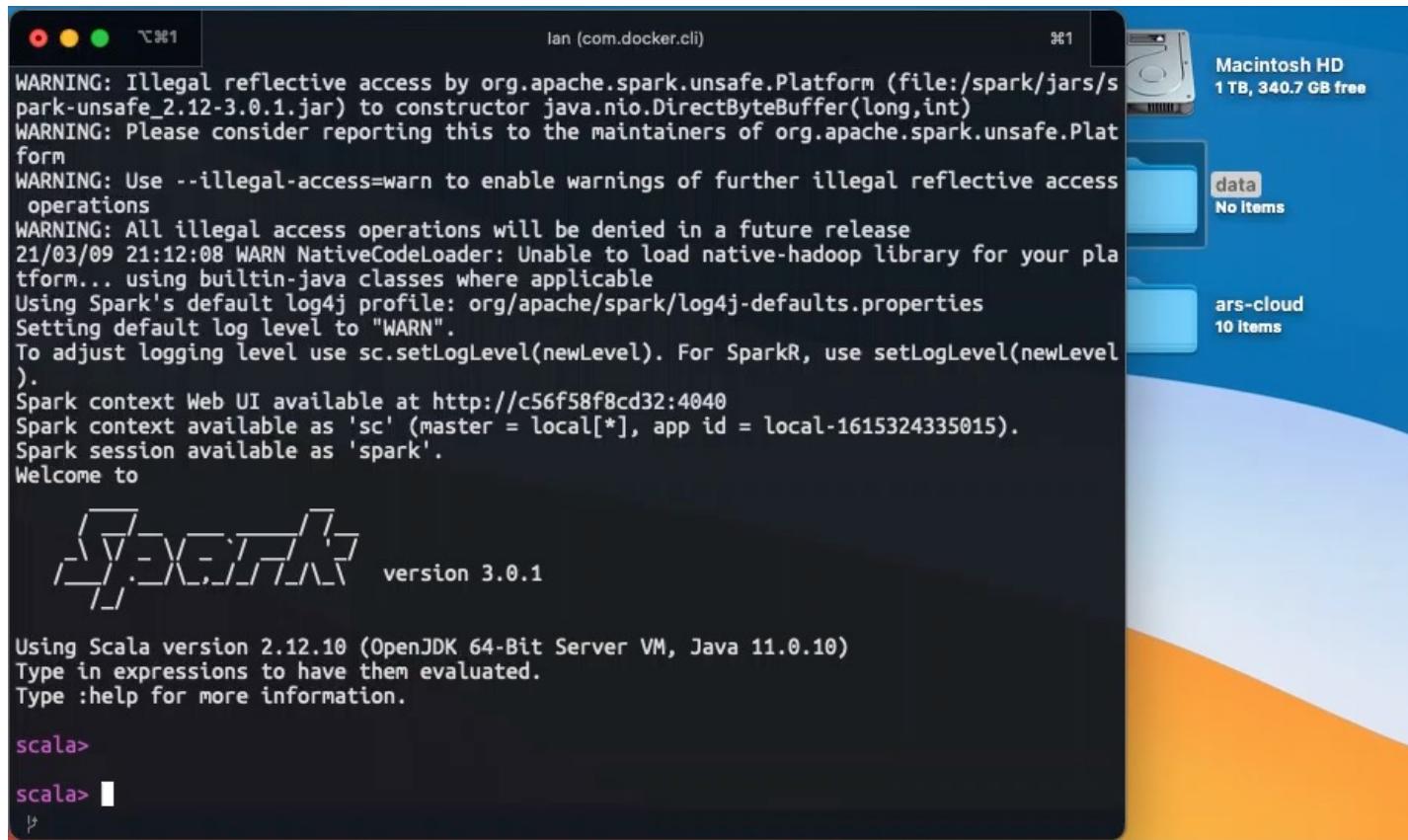
Script

```
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!



A screenshot of a Mac desktop environment. On the left, a terminal window titled "ian (com.docker.cli)" displays the startup logs for a Spark application. On the right, a file browser window shows the contents of the "data" folder on "Macintosh HD".

```
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 builtin-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://c56f58f8cd32:4040
Spark context available as 'sc' (master = local[*], app id = local-1615324335015).
Spark session available as 'spark'.
Welcome to
    \_____
   /       \
  /  \  .  /
 /  _ \ \  /
 /  / \ \_ \
 /  / \ \_ \
version 3.0.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
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

```
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.

Date Filter (full or partial) → .keepDate

```
Dates: val dates =  
List("2008","200908","20070502")
```

Example:

```
.keepDate(List("200804"),  
ExtractDate.DateComponent.YYYYMM)
```

```
.keepDate(List("2008","2015"),  
ExtractDate.DateComponent.YYYY)
```

URLs → .keepUrlPatterns

```
URLs: val urls = Set("archive.org","uwaterloo.ca","yorku.ca")
```

Example:

```
.keepUrlPatterns(Set("www.davidsuzuki.org".r))
```

Language → .keepLanguages

```
Languages: val languages = Set("en")
```

Example:

```
.keepLanguages(Set("en"))
```

Uses the
ISO 639.2
language
codes

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 (creates a .gexf output file)

```
import io.archivesunleashed._  
import io.archivesunleashed.udfs._  
import io.archivesunleashed.app._  
  
val graph = RecordLoader.loadArchives("aut-  
resources/Sample-Data/*.gz", sc)  
    .webgraph.groupBy(  
        $"crawl_date",  
  
removePrefixWWW(extractDomain($"src")).as("src_domain"),  
  
removePrefixWWW(extractDomain($"dest")).as("dest_domain"))  
    .count()  
    .filter(!($"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

```
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

```
links: Array[(String, Int)] = Array((http://www.liberal.ca/shared/images/logo_footer.png,1968),  
(http://www.liberal.ca/images/pages/graphics/share_e.gif,1966), (http://www.gca.ca/indexcms/in-  
gleer.gif,1780), (http://www.liberal.ca/images/pages/features/liberaltv_e.png,1116), (http://w-  
ww.liberal.ca/images/section-headers/get-involved.png,1114), (http://www.plaxo.com/images/abc/b-  
uttons/add_button.gif,1114), (http://www.liberal.ca/images/section-headers/newsroom.png,854),  
(http://www.davidsuzuki.org/files/dent.gif,764), (http://i.ytimg.com/vi/8HeuyC3ysA/default.jpg,  
493), (http://www.fairvote.ca/sites/fairvote.ca/themes/fvc_ruby/logo.png,465))
```

Archives Unleashed Project

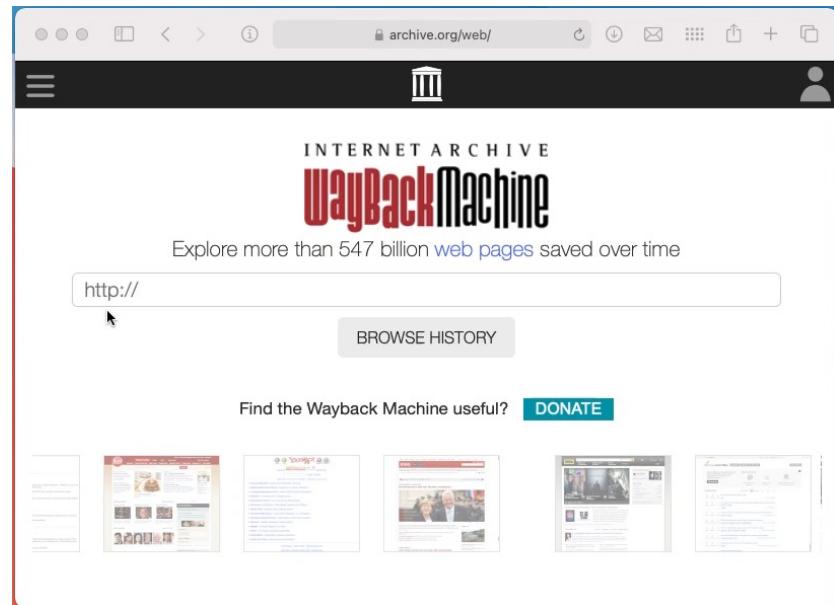
- Collections Analytics
 - List of URLs (some or all)
- Plain Text Extraction
 - All plain text
 - Plain text by domain
- Analysis of Site Link Structure
 - Exporting to Gephi Directly
- **Image Analysis**
 - **Most frequent image URLs in a collection**
- Example using DF (Dataframes)
 - Top Domains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



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

```
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)
```

DataFrame Output

Domain	count
www.equalvoice.ca	4644
www.liberal.ca	1968
greenparty.ca	732
www.policyalterna...	601
www.fairvote.ca	465
www.ndp.ca	417
www.davidsuzuki.org	396
www.canadiancrc.com	90
www.gca.ca	40
communist-party.ca	39
westernblockparty...	26
www.nosharia.com	24
canadianactionpar...	22

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

- Example using DF (DataFrames)
 - Top Domains
 - Image Analysis

Script

```
import io.archivesunleashed._  
import io.archivesunleashed.udfs._  
  
val df = RecordLoader.loadArchives("/aut-  
resources/Sample-Data/*.gz",  
sc).images();  
  
df.select($"url", $"filename",  
$"extension", $"mime_type_web_server",  
$"mime_type_tika", $"width", $"height",  
$"md5", $"sha1", $"bytes")  
.orderBy(desc("md5"))  
.show()
```

DataFrame Output

	url	filename	extension	mime_type_web_server	mime_type_tika	width	height	md5	bytes
1	http://agoracosmo... Valerie_Armstrong...	.jpg	image/jpeg	image/jpeg	109	127	f4f167c12dd52506...	/9j/4AAQSKZJRgABA...	
2	http://www.fin.gc...	taxes_e.gif	gif	image/gif	image/gif	132	44	fe93eaa2d1346c488...	R01G0D1hhAAsPCAA...
3	http://www.herita...	u7f.jpg	jpg	image/jpeg	image/jpeg	198	272	fe5c459dee1de758a...	/9j/4AAQSKZJRgABA...
4	http://www.davids...	Challengeshirt2.jpg	jpg	image/jpeg	image/jpeg	216	249	fe1a5ee5946d91a...	/9j/4AAQSKZJRgABA...
5	http://agoracosmo...	budget_e.gif	gif	image/gif	image/gif	132	44	fd2be109231b9ca5...	R01G0D1hhAAsPCAA...
6	http://agoracosmo...	Alexander_Pappas.jpg	jpg	image/jpeg	image/jpeg	109	137	fbff272a59bc5623...	/9j/4AAQSKZJRgABA...
7	http://agoracosmo...	Agnes_Sroczynski.jpg	jpg	image/jpeg	image/jpeg	109	105	fb8b34b64825df699...	/9j/4AAQSKZJRgABA...
8	http://partimarij...	slive_top_yellow...	gif	image/gif	image/gif	200	15	fb4e7ab247dcce36...	R01G0D1hyAAQAJECA...
9	http://blocquebec...	manufacture_1.gif	gif	image/gif	image/gif	545	21	f83lea53a382ef06...	R01G0D1hhQWPACAA...
10	http://coat.ncf.c...	smith_a.jpg	jpg	image/jpeg	image/jpeg	211	316	f74e58e4d894d7825...	/9j/4AAQSKZJRgABA...
11	http://www.pm.gc...	lhs_sub_0.jpg	jpg	image/jpeg	image/jpeg	140	20	f6d8513ffd5b897f...	/9j/4AAQSKZJRgABA...
12	http://coat.ncf.c...	dodd.jpg	jpg	image/jpeg	image/jpeg	286	391	f49d325880a4b47d7...	/9j/4AAQSKZJRgABA...
13	http://www.ccsd.c...	photo.jpg	jpg	image/jpeg	image/jpeg	160	208	f43cae2293a19156e...	/9j/4AAQSKZJRgABA...
14	http://www.caanda...	p12.jpg	jpg	image/jpeg	image/jpeg	91	91	f12e389214b7-2fa5...	/9j/4AAQSKZJRgABA...

Archives Unleashed Toolkit

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

<https://aut.docs.archivesunleashed.org>

The screenshot shows the official documentation site for the Archives Unleashed Toolkit. At the top right, there are links for Docs, Project, GitHub, and News. Below that is a "Table of Contents" sidebar with sections like Getting Started, Generating Results, Filtering Results, Standard Derivatives, What to do with Results, and Citing Archives Unleashed. The main content area features a header "Archives Unleashed Toolkit 0.90.0" with a small icon. It includes a "Home" section with a "The Toolkit" link, a "Getting Started" section with links to Dependencies, Usage, The Toolkit at Scale, DataFrame Schemas, and Toolkit Walkthrough, and a "Generating Results" section with links to Collection Analysis, Text Analysis, Link Analysis, Binary Analysis, and Filtering Results. Under Filtering Results, there are links for RDD Filters and DataFrame Filters. The Standard Derivatives section has links for The Toolkit with spark-submit, AU Cloud Scholarly Derivatives, Extract Binary Info, and Extract Binaries to Disk. The "What to do with Results" section includes links for DataFrame Results and RDD Results. A sidebar on the right contains a summary of the toolkit's capabilities and links to further reading and acknowledgments.

Archives Unleashed Toolkit 0.90.0

Docs Project GitHub News

Table of Contents

- Getting Started
- Generating Results
- Filtering Results
- Standard Derivatives
- What to do with Results
- Citing Archives Unleashed
- Further Reading
- Acknowledgments

Home

[The Toolkit](#)

Getting Started

[Dependencies](#)
[Usage](#)
[The Toolkit at Scale](#)
[DataFrame Schemas](#)
[Toolkit Walkthrough](#)

Generating Results

[Collection Analysis](#)
[Text Analysis](#)
[Link Analysis](#)
[Binary Analysis](#)

Filtering Results

[RDD Filters](#)
[DataFrame Filters](#)

Standard Derivatives

[The Toolkit with spark-submit](#)
[AU Cloud Scholarly Derivatives](#)
[Extract Binary Info](#)
[Extract Binaries to Disk](#)

What to do with Results

[DataFrame Results](#)
[RDD Results](#)

The Toolkit

The Archives Unleashed Toolkit is an open-source platform for analyzing web archives built on Apache Spark, which provides powerful tools for analytics and data processing.

This documentation is based on a cookbook approach, providing a series of "recipes" for addressing a number of common analytics tasks to provide inspiration for your own analysis. We generally provide examples for **resilient distributed datasets (RDD)** in Scala, and **DataFrames** in both Scala and Python. We leave it up to you to choose Scala or Python flavours of Spark.

If you want to learn more about Apache Spark, we highly recommend [Spark: The Definitive Guide](#).

Table of Contents

Our documentation is divided into several main sections, which cover the Archives Unleashed Toolkit workflow from analyzing collections to understanding and working with the results.

Getting Started

- [Dependencies](#)
- [Usage](#)
- [Using the Archives Unleashed Toolkit at Scale](#)
- [Toolkit Walkthrough](#)
- [DataFrame Schemas](#)

Generating Results

- [Collection Analysis: How do I...](#)
 - [Extract All URLs](#)
 - [Extract Top-Level Domains](#)
 - [Extract Different Subdomains](#)
 - [Extract HTTP Status Codes](#)
 - [Extract the Location of the Resource in ARCs and WARCs](#)

Hands-on: External Tools

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.

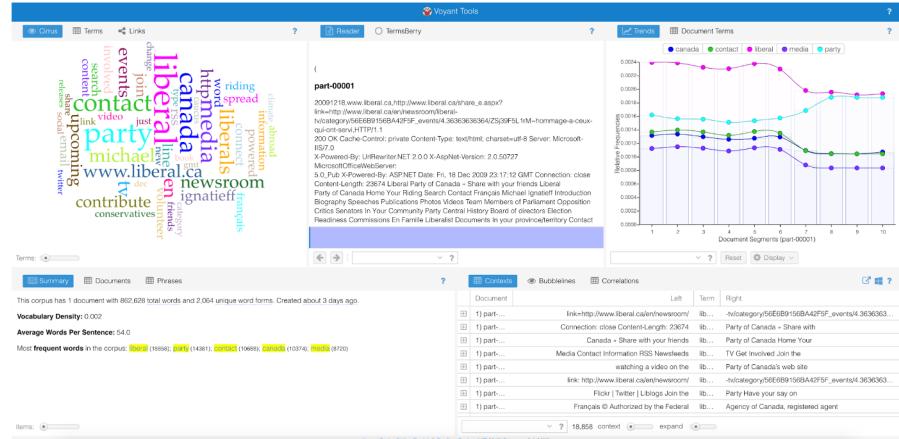
Voyant Tools: voyant-tools.org



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

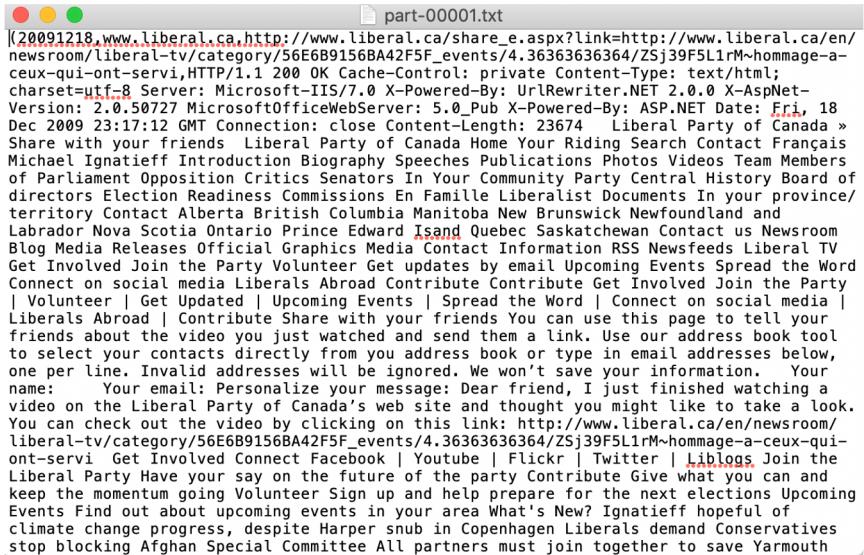


Analyzing Extracted Text

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.

```
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(r.getContentString)))  
  .saveAsTextFile("/data/liberal-party-text")
```



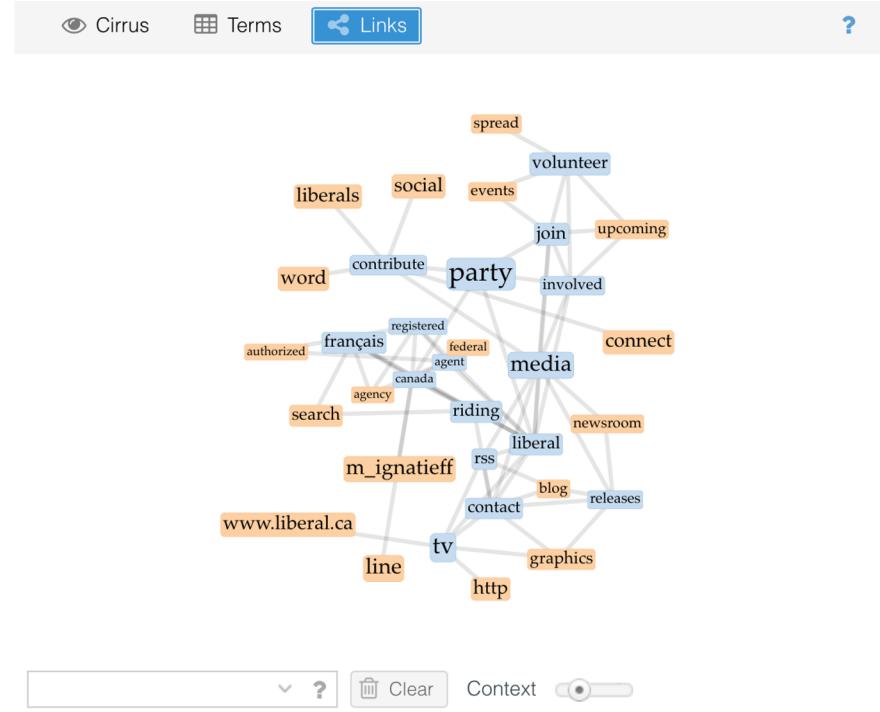
The screenshot shows a terminal window with a gray background. At the top, there are three small colored circles (red, yellow, green) followed by the file name "part-00001.txt". The main area of the terminal contains a large amount of text, which is a direct dump of the extracted content from the liberal.ca website. The text is mostly black with some blue and red links visible. It includes various sections of the website's content, such as news articles, sidebar links, and social media integration code.

Voyant Tools

Links

Trends
Contexts

Voyant's Links tool allows you to visualize context relationships between frequently-used keywords.



Voyant Tools

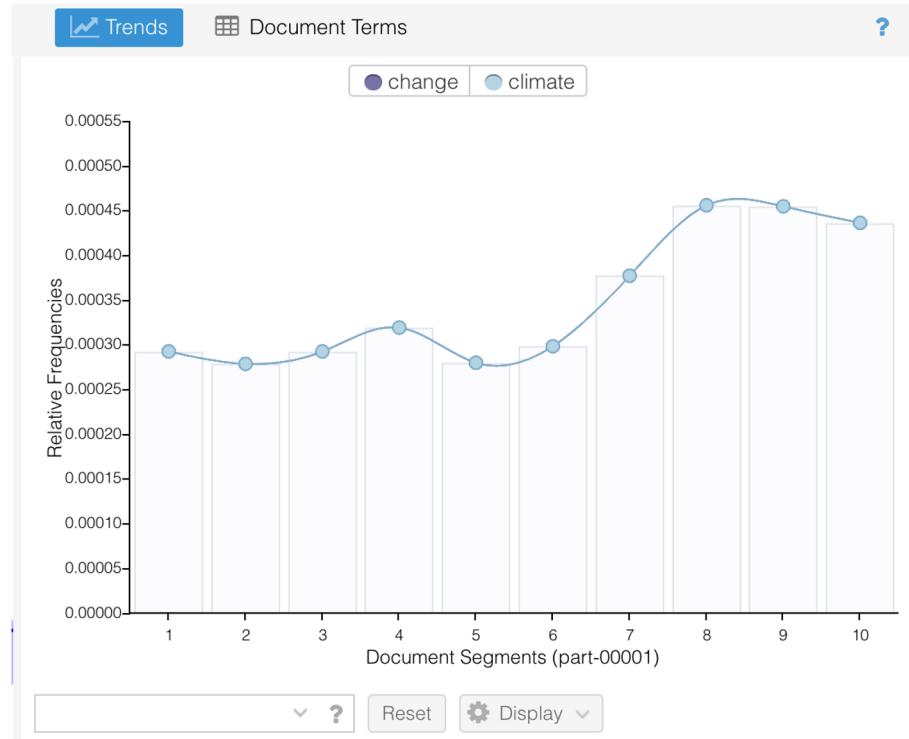


Links

Trends

Contexts

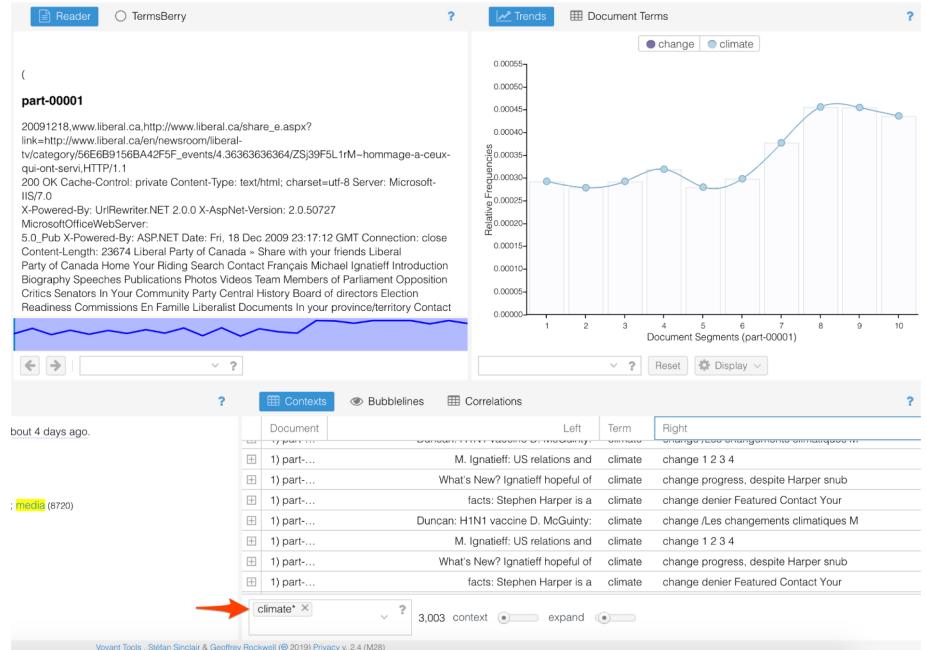
Voyant's Trends tool allows you to graph the frequency of a keyword throughout your text file.



Voyant Tools

Links
Trends
Contexts

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.



VOYANT

see through your text

Add Texts

Type in one or more URLs on separate lines or paste in a full text.

 Open  Upload  Reveal

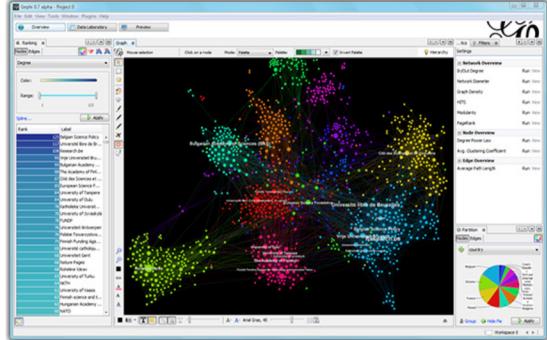
Voyant Tools is a web-based reading and analysis environment for digital texts.

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



The Open Graph Viz Platform

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Runs on Windows, Mac OS X and Linux.

[Learn More on Gephi Platform »](#)

[Download FREE Gephi 0.9.2](#)

[Release Notes](#) | [System Requirements](#)

[Features](#) [Screenshots](#)
[Quick start](#) [Videos](#)

Support us! We are **non-profit**. Help us to **innovate** and **empower** the community by donating only 8€:

[Donate](#)



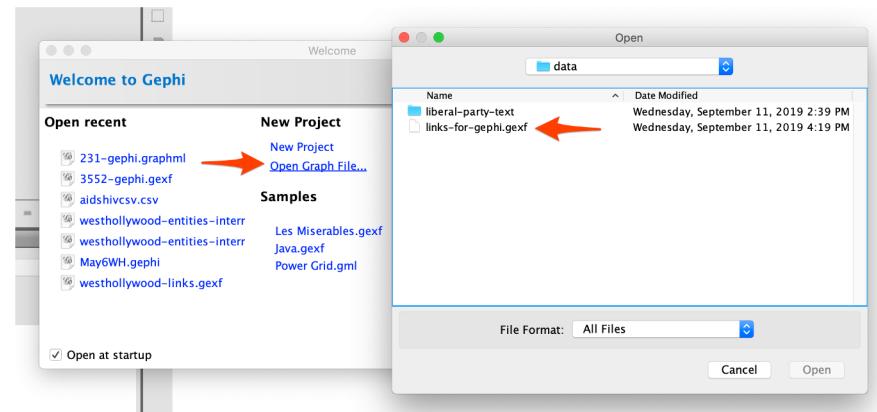
Gephi

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.

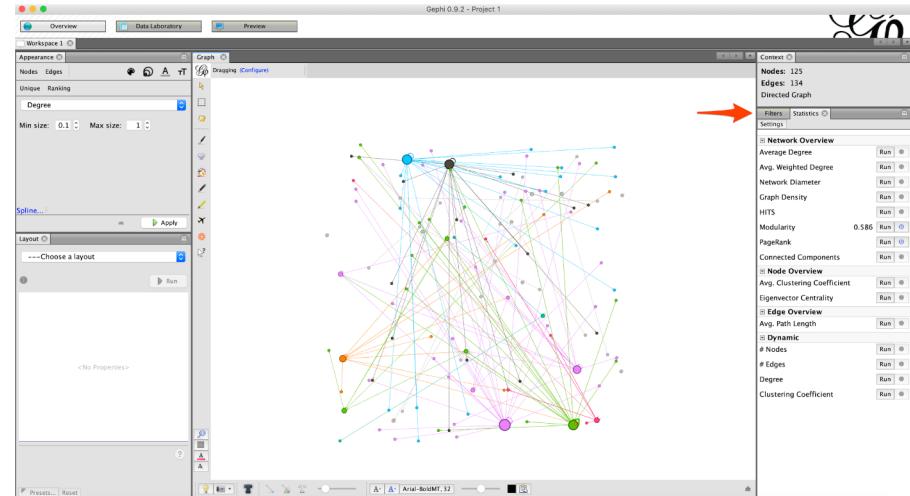
```
import io.archivesunleashed._  
import io.archivesunleashed.app._  
import io.archivesunleashed.matchbox._  
  
val links = RecordLoader.loadArchives("/aut-resources/Sample-Data/*.gz", sc)  
    .keepValidPages()  
    .map(r => (r.getCrawlDate, ExtractLinks(r.getUrl, r.getContentString)))  
    .flatMap(r => r._2.map(f => (r._1, ExtractDomain(f._1).replaceAll("^\\s*www\\.", "")  
        .filter(r => r._2 != "" && r._3 != "")  
        .countItems())  
        .filter(r => r._2 > 5))  
  
WriteGEXF(links, "/data/links-for-gephi.gexf")
```



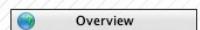
Gephi

Open source visualization and exploration software.

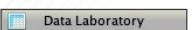
Step 2: Use Statistics and Filters tools to organize data.



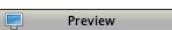
Archives Unleashed Learning Guide:
Network Graphing Archived Websites
with Gephi
<https://cloud.archivesunleashed.org/derivatives/gephi>



Overview



Data Laboratory



Preview



Appearance **Graph** **Context**

Nodes Edges Dragging (Configure)

Welcome

Welcome to Gephi

Open recent

- test.gexf
- gephic.csv

New Project

- New Project
- Open Graph File...

Samples

- Les Miserables.gexf
- Java.gexf
- Power Grid.gml

Open at startup

<No Properties>

Layout ---Choose a layout---

Presets... Reset

T T A A Arial-BoldMT, 32

Network Overview
Average Degree
Avg. Weighted Degree
Network Diameter
Graph Density
HITS
Modularity
PageRank
Connected Components
Node Overview
Avg. Clustering Coefficient
Eigenvector Centrality
Edge Overview
Avg. Path Length
Dynamic
Nodes
Edges
Degree
Clustering Coefficient

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](https://twitter.com/unleasharchives)

YouTube [UC4Sq0Xi6UWhYK2VbmAzFhAw](https://www.youtube.com/channel/UC4Sq0Xi6UWhYK2VbmAzFhAw)

- 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

Jacquelyn Bulao. How Much Data Is Created Every Day in 2020? March 18, 2021. TechJury.

<https://techjury.net/blog/how-much-data-is-created-every-day/#gref>

Joseph Johnson. Global Digital Population as of January 2021. March 5, 2021. Statista.

<https://www.statista.com/statistics/617136/digital-population-worldwide/>

Images

Nick Ruest and Toke Eskildesen. #WomensMarch.

Created via Juxta. <https://ruebot.net/visualizations/wm/>

Everyday basics on Unsplash

Robynne Hu on Unsplash

Gordon Johnson from Pixabay

Erik Mclean on Unsplash

Jason Leung on Unsplash

Sources

Software Mentioned

Docker	https://www.docker.com
WayBack Machine	https://archive.org/web/
Voyant Tools	https://voyant-tools.org
Gephi	https://gephi.org

Example Dataset

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