



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



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



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



NATIONAL
LIBRARY
OF AUSTRALIA

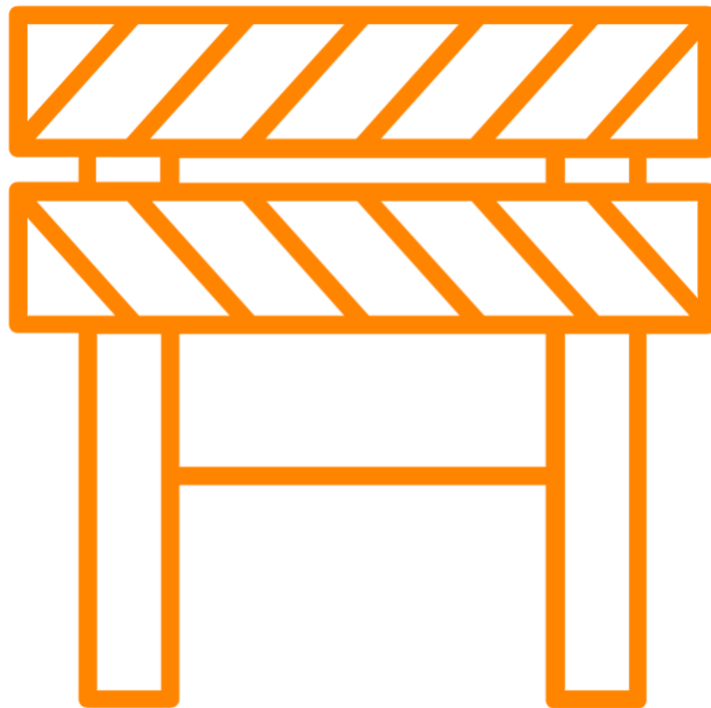
2021 Petabytes of data for studying topics from the 1990s forward



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.

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
Warlight
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 Project

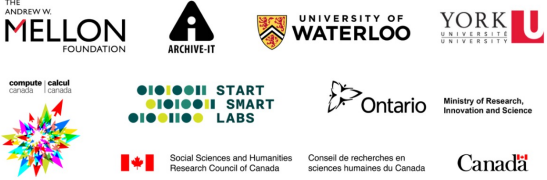
The Archives Unleashed project aims to make petabytes of historical internet content accessible to scholars and others interested in researching the recent past. Our team develops web archive search and data analysis tools to enable scholars, librarians and archivists to access, share, and investigate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



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 Toolkit Workshop

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 are navigation links: HOME, EXPLORE, LEARN MORE, and CONTACT US. The Archive-It logo is on the left, and social media icons and a 'Login' button are on the right. Below the navigation, the breadcrumb trail reads: Explore >> University of Toronto >> Canadian Political Parties and Political Interest Groups.

The main content area features a green header for the collection: 'Canadian Political Parties and Political Interest Groups', collected by the University of Toronto. It includes metadata: 'Archived since: Oct. 2005', a description of the collection, subject tags ('Politics & Elections'), rights information, and the collector name.

Below this is a 'Narrow Your Results' section with filters for 'Group' and 'Subject', each with a 'Sort By: Count' and '(A-Z)' option. A search bar is present with 'Enter search terms here' and 'Search' and 'Clear' buttons. The page indicates 'Page 1 of 2 (125 Total Results)' and has a 'Next Page' button.

The results list includes:

- Group:** Quarterly Crawl Brozzler (1)
- Subject:** Conservative Party of Canada (4), New Democratic Party of Canada (4), Essor National Parti du Canada (3), Les Intellectuels Pour la Souveraineté du Québec (3), Liberal Party of Canada (3). A 'More' button is visible.
- Creator:** Alex E.H. Ng (1), Canada First Immigration Reform Committee (1), Canadian Public Health Association (CPHA) (1), Environmental Defence Canada (1), National Citizens Alliance (1). A 'More' button is visible.

Two detailed result entries are shown:

- Title:** Cosmopolitan Party of Canada
URL: <http://agoracosmopolite.com/>
Description: Renamed the Progressive Nationalist Party of Canada in 2007. Captured 59 times between Oct 4, 2005 and Nov 3, 2012
Subject: Cosmopolitan Party of Canada
- Title:** Bloc Québécois
URL: <http://blocquebecois.org/>
Description: Captured 68 times between Feb 1, 2009 and Jan 30, 2021
Subject: Bloc Québécois
Rights: Bloc Québécois

A third result entry is partially visible: **Title:** Canada First Immigration Reform Committee

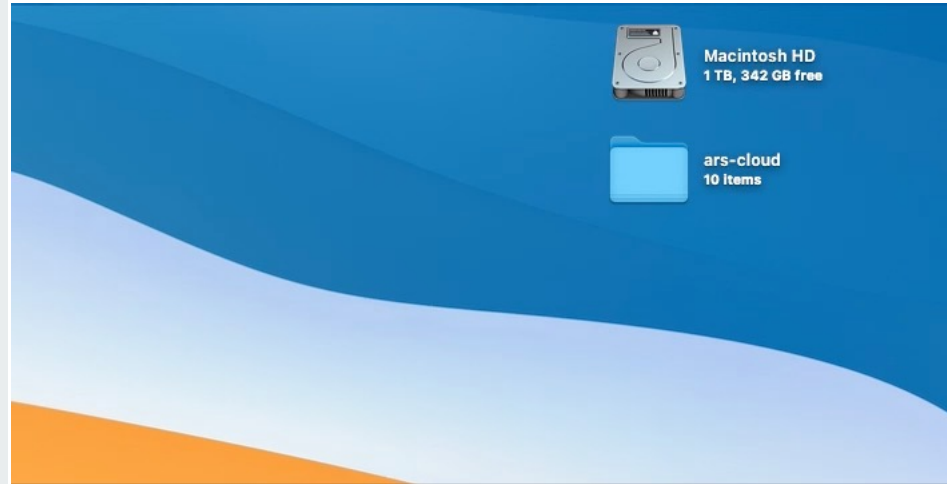
<https://archive-it.org/collections/227>

How We'll Do It



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

Like so!



Setup Docker

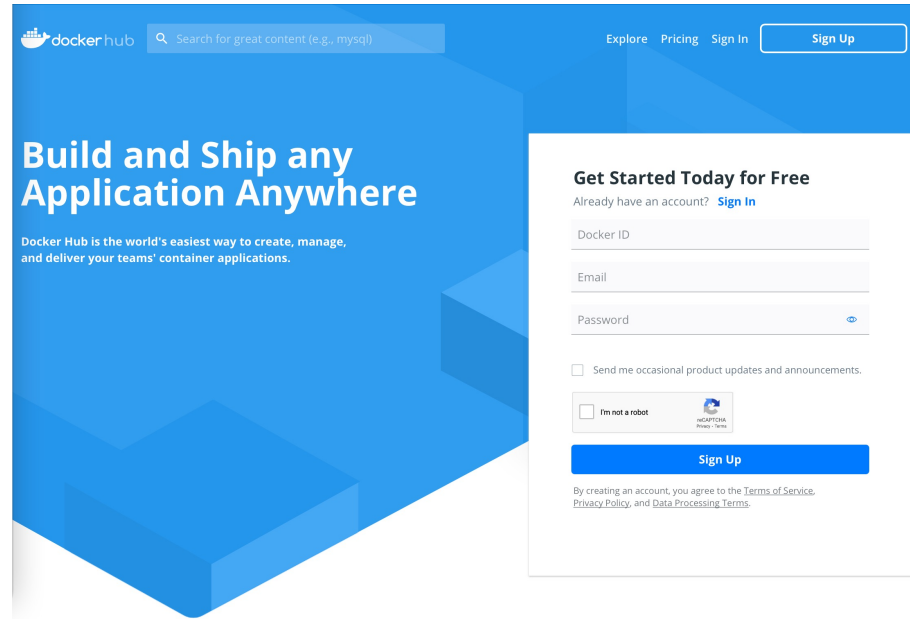
Setup Docker

“**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.”

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

Step 1: Sign Up for free Docker ID

The image shows the Docker Hub website's sign-up page. The header features the Docker Hub logo, a search bar with the placeholder text "Search for great content (e.g., mysql)", and navigation links for "Explore", "Pricing", "Sign In", and a "Sign Up" button. The main content area has a blue background with the text "Build and Ship any Application Anywhere" and a sub-headline: "Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications." On the right side, there is a white sign-up form titled "Get Started Today for Free". The form includes a "Sign In" link for existing users, input fields for "Docker ID", "Email", and "Password", a checkbox for "Send me occasional product updates and announcements.", and a checkbox for "I'm not a robot" with a CAPTCHA icon. A blue "Sign Up" button is at the bottom of the form, followed by a link to the "Terms of Service, Privacy Policy, and Data Processing Terms".

<https://hub.docker.com>

Setup Docker

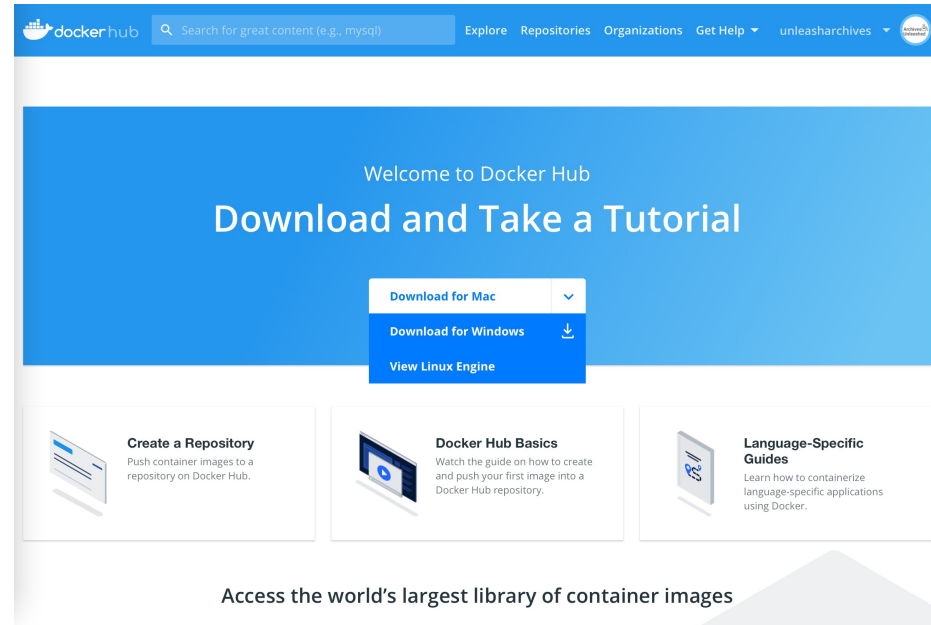
“**Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers.

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.”

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

Step 2: Login to Docker + Select OS System

<https://www.docker.com>



The screenshot shows the Docker Hub website homepage. At the top, there is a blue navigation bar with the Docker Hub logo, a search bar, and links for Explore, Repositories, Organizations, Get Help, and unleasharchives. Below the navigation bar, a large blue banner reads "Welcome to Docker Hub" and "Download and Take a Tutorial". Underneath the banner, there are three main sections: "Create a Repository" (with a subtext "Push container images to a repository on Docker Hub."), "Docker Hub Basics" (with a subtext "Watch the guide on how to create and push your first image into a Docker Hub repository."), and "Language-Specific Guides" (with a subtext "Learn how to containerize language-specific applications using Docker."). At the bottom of the page, there is a grey banner that says "Access the world's largest library of container images".

Setup Docker

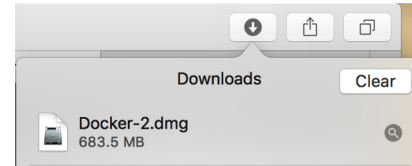
“**Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers.

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.”

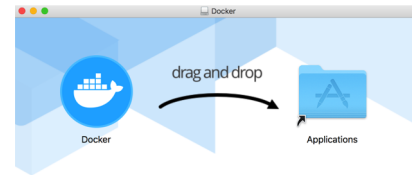
Citation: “What is Docker” <https://opensource.com/resources/what-docker>

Step 3: Run through Docker install

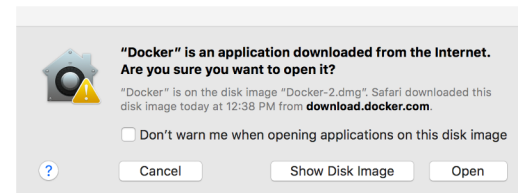
1. Double click .dmg folder to launch install



2. Drag and drop into Applications folder



3. Open Docker **NOTE:** Docker may require access depending on system requirements



Setup Docker

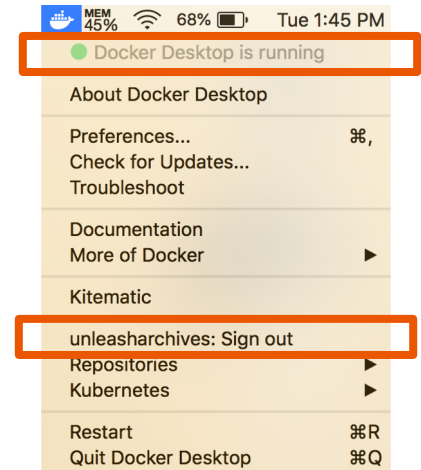
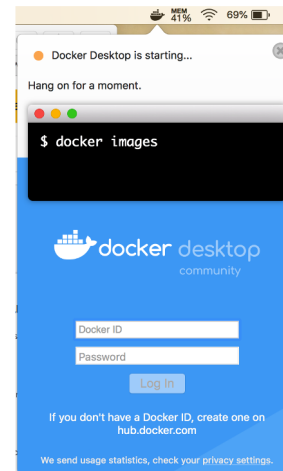
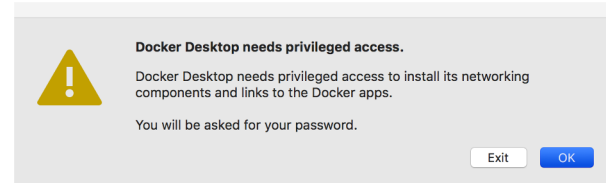
“**Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers.

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>

Step 3: Run through Docker install (cont.)

Open Docker - green dot indicates Docker is running



Setup Docker

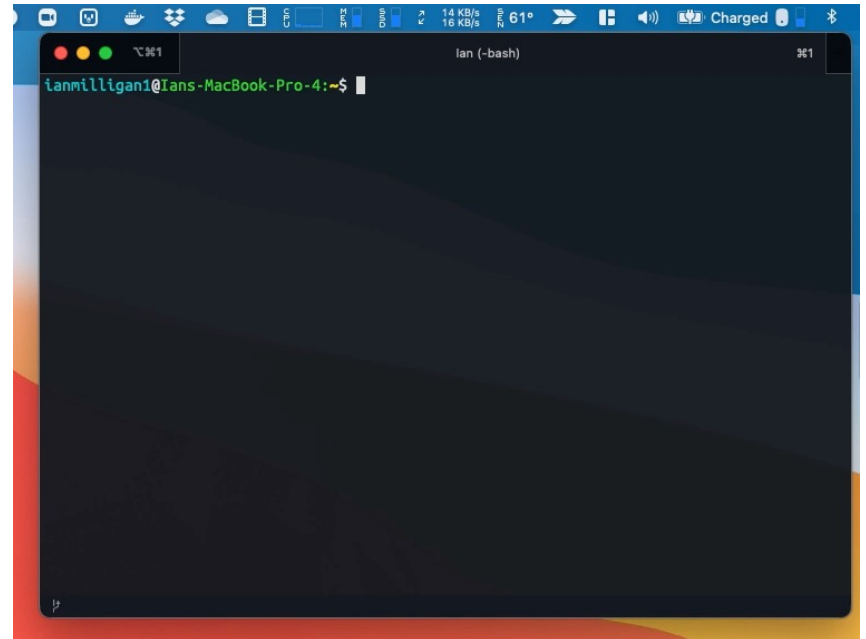
“**Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers.

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.”

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

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

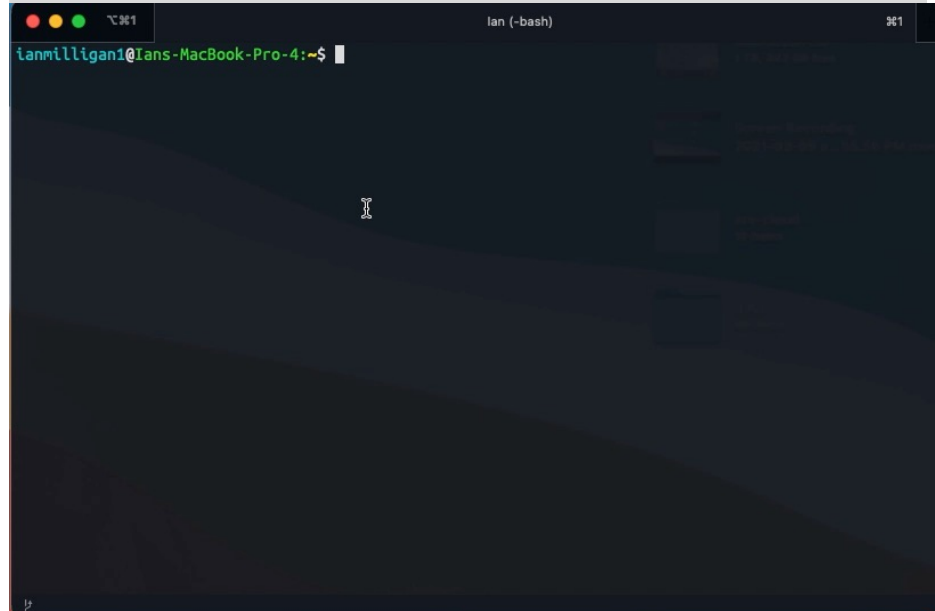
<https://aut.docs.archivesunleashed.org>

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 WARC

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
```

```
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/s  
park-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.Plat  
form  
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 pla  
tform... 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  
  
SPARK 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> 
```

Archives Unleashed Project

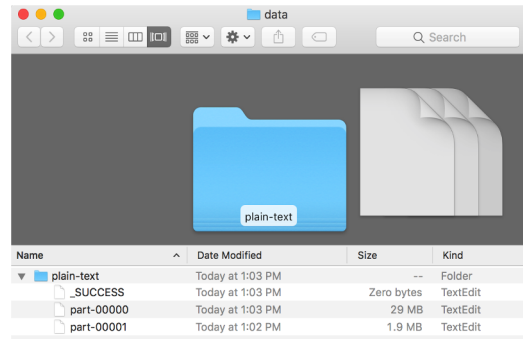
- Collections Analytics
 - List of URLs (some or all)
- **Plain Text Extraction**
 - **All plain text**
 - Plain text by domain
- Analysis of Site Link Structure
 - Exporting to Gephi Directly
- Image Analysis
 - Most frequent image URLs in a collection
- Example using DF (Dataframes)
 - Top Domains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

Script

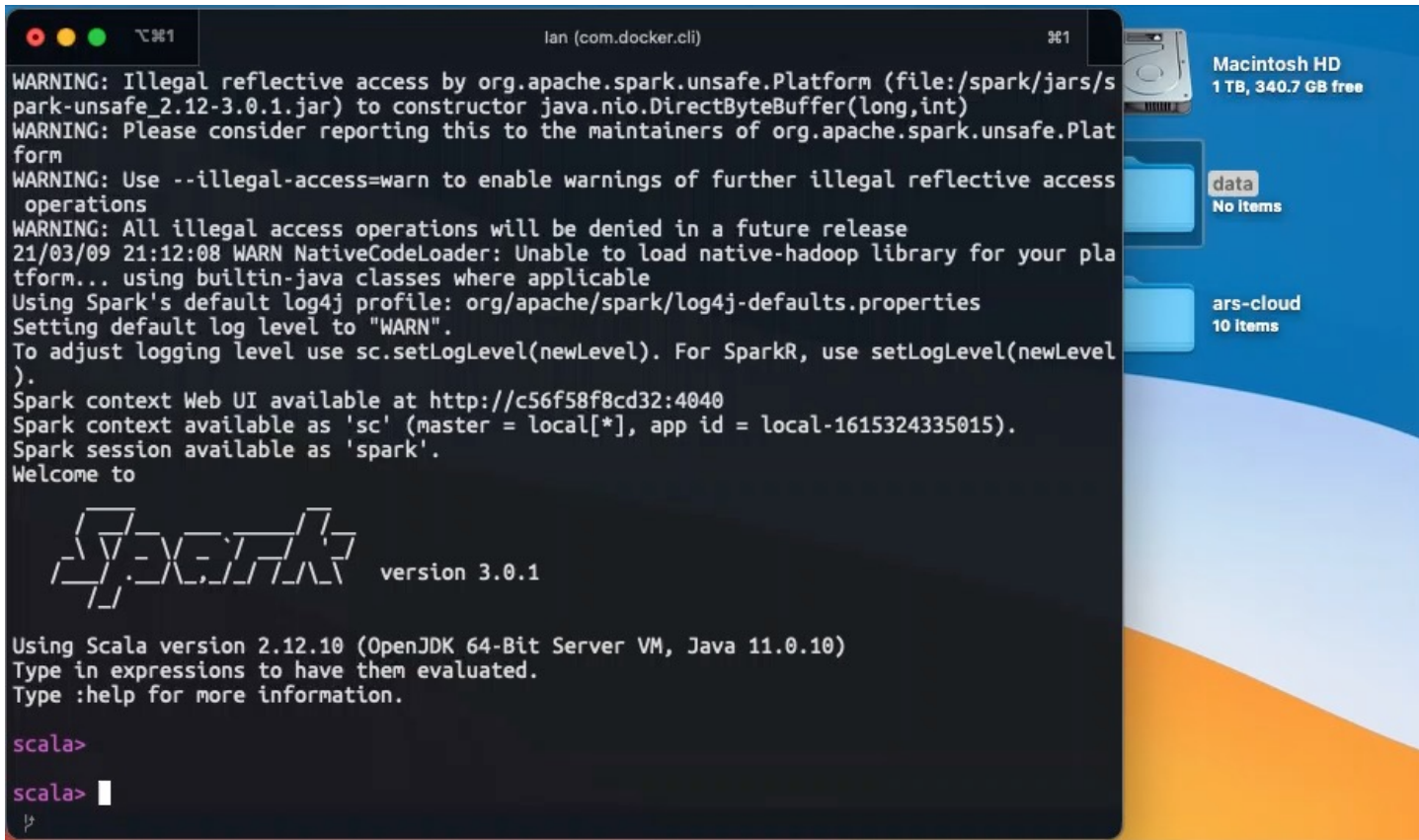
```
import io.archivesunleashed._
import io.archivesunleashed.matchbox._

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!



The image shows a terminal window with a dark background and a file manager sidebar on the right. The terminal output displays the following text:

```
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> |
```

The file manager sidebar on the right shows the following items:

- Macintosh HD: 1 TB, 340.7 GB free
- data: No Items
- ars-cloud: 10 Items

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:

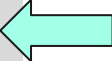
```
.keepUrlPatterns(Set("www.davidsuzuki.org".r))
```

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/im
g/leer.gif,1780), (http://www.liberal.ca/images/pages/features/liberal_tv_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/8HeuyBC3ysA/default.jpg,
493), (http://www.fairvote.ca/sites/fairvote.ca/themes/fvc_ruby/logo.png,465))
```


Archives Unleashed Project

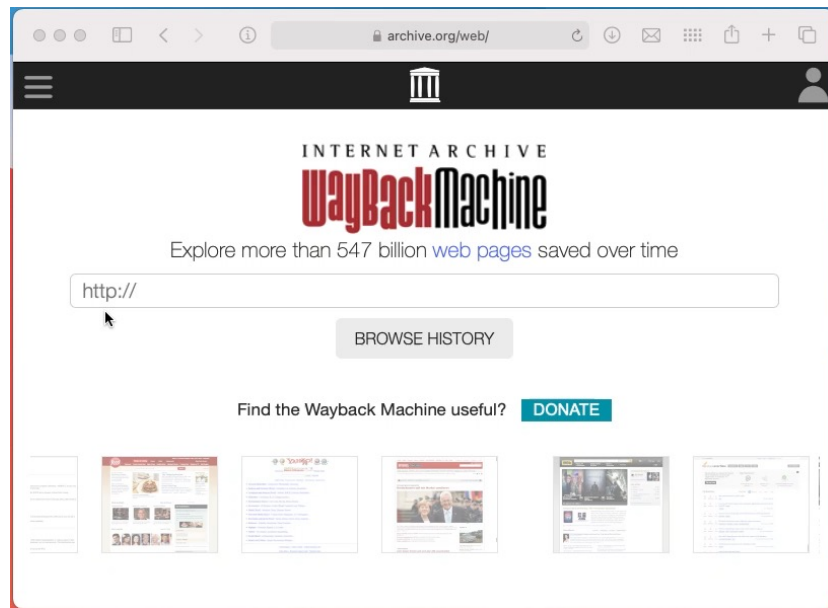
- Collections Analytics
 - List of URLs (some or all)
- Plain Text Extraction
 - All plain text
 - Plain text by domain
- Analysis of Site Link Structure
 - Exporting to Gephi Directly
- **Image Analysis**
 - **Most frequent image URLs in a collection**
- Example using DF (Dataframes)
 - Top Domains
 - 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|
| canadianactionar...| 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 collection
- **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
http://agoracosmo...	Valerie_Armstrong...	jpg	image/jpeg	image/jpeg	109	127	ff4f167c12dd52586...	/9j/4AAQSkZJRgABA...
http://www.fin.gc...	taxes_e.gif	gif	image/gif	image/gif	132	44	fe93eaa2d1346c488...	R01G0D1hhAAAsPcAA...
http://www.herita...	uf7.jpg	jpg	image/jpeg	image/jpeg	198	272	fe5c459dee1de758a...	/9j/4AAQSkZJRgABA...
http://www.davids...	Challengeshirt2.jpg	jpg	image/jpeg	image/jpeg	216	249	fe16a5ee5946b91a...	/9j/4AAQSkZJRgABA...
http://www.fin.gc...	budget_e.gif	gif	image/gif	image/gif	132	44	fd2be1089231b9ca5...	R01G0D1hhAAAsPcAA...
http://agoracosmo...	Alexander_Pappas.jpg	jpg	image/jpeg	image/jpeg	109	137	fbf0272a650bc5623...	/9j/4AAQSkZJRgABA...
http://agoracosmo...	Agnes_Sroczyński.jpg	jpg	image/jpeg	image/jpeg	109	105	fb8b34b647dccb36...	/9j/4AAQSkZJRgABA...
http://partimarij...	slice_top_yellow...	gif	image/gif	image/gif	200	16	fb4e7ab247dccb36...	R01G0D1hYAAQAJECA...
http://bloquebec...	manufacture_1.gif	gif	image/gif	image/gif	545	22	fb31ea53a3822ef06...	R01G0D1hYAAQAJECA...
http://coat.ncf.c...	smith_a.jpg	jpg	image/jpeg	image/jpeg	211	316	f74e58e4d894d7825...	/9j/4AAQSkZJRgABA...
http://www.pm.gc...	lhs_sub_0.jpg	jpg	image/jpeg	image/jpeg	140	28	f6d8513ffdc58b97f...	/9j/4AAQSkZJRgABA...
http://coat.ncf.c...	dodd.jpg	jpg	image/jpeg	image/jpeg	286	391	f69d325809a4b47d...	/9j/4AAQSkZJRgABA...
http://www.ccsd.c...	photo.jpg	jpg	image/jpeg	image/jpeg	160	280	f43cae2293a19156e...	/9j/4AAQSkZJRgABA...
http://www.ccsd.c...	nl10	image	image/jpeg	image/jpeg	8	8	f13c382214b73fae...	/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>

Archives Unleashed Toolkit **0.90.0** [Docs](#) [Project](#) [GitHub](#) [News](#)

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](#)

Table of Contents

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

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



Add Texts 🔍 🔗 ?

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

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

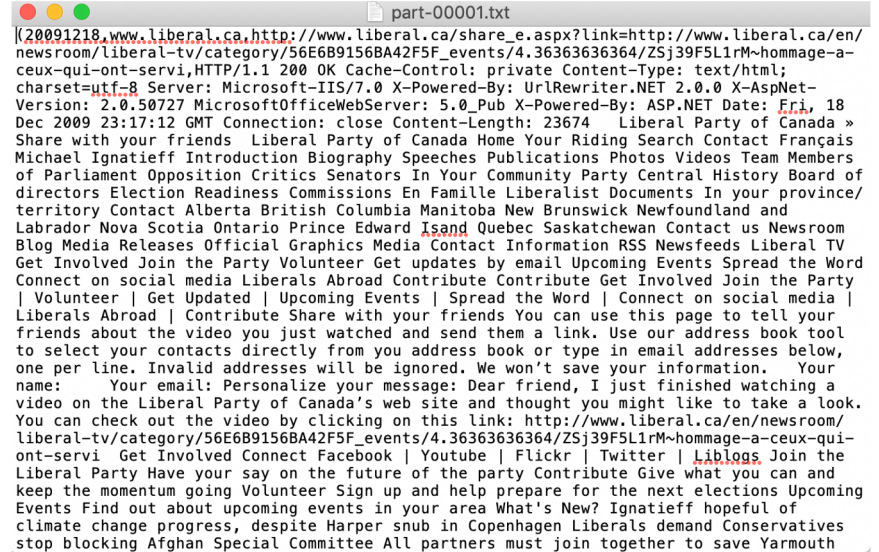
Analyzing Extracted Text

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")
```



part-00001.txt

```
(20091218, www.liberal.ca, http://www.liberal.ca/share_e.aspx?link=http://www.liberal.ca/en/newsroom/liberal-tv/category/56E6B9156BA42F5F_events/4.36363636364/Z5j39F5L1rM~homage-a-ceux-qui-ont-servi, HTTP/1.1 200 OK Cache-Control: private Content-Type: text/html; charset=utf-8 Server: Microsoft-IIS/7.0 X-Powered-By: UrlRewriter.NET 2.0.0 X-AspNet-Version: 2.0.50727 MicrosoftOfficeWebServer: 5.0_Pub X-Powered-By: ASP.NET Date: Fri, 18 Dec 2009 23:17:12 GMT Connection: close Content-Length: 23674 Liberal Party of Canada » Share with your friends Liberal Party of Canada Home Your Riding Search Contact Français Michael Ignatieff Introduction Biography Speeches Publications Photos Videos Team Members of Parliament Opposition Critics Senators In Your Community Party Central History Board of directors Election Readiness Commissions En Famille Liberalist Documents In your province/territory Contact Alberta British Columbia Manitoba New Brunswick Newfoundland and Labrador Nova Scotia Ontario Prince Edward Island Quebec Saskatchewan Contact us Newsroom Blog Media Releases Official Graphics Media Contact Information RSS Newsfeeds Liberal TV Get Involved Join the Party Volunteer Get updates by email Upcoming Events Spread the Word Connect on social media Liberals Abroad Contribute Contribute Get Involved Join the Party | Volunteer | Get Updated | Upcoming Events | Spread the Word | Connect on social media | Liberals Abroad | Contribute Share with your friends You can use this page to tell your friends about the video you just watched and send them a link. Use our address book tool to select your contacts directly from you address book or type in email addresses below, one per line. Invalid addresses will be ignored. We won't save your information. Your name: Your email: Personalize your message: Dear friend, I just finished watching a video on the Liberal Party of Canada's web site and thought you might like to take a look. You can check out the video by clicking on this link: http://www.liberal.ca/en/newsroom/liberal-tv/category/56E6B9156BA42F5F_events/4.36363636364/Z5j39F5L1rM~homage-a-ceux-qui-ont-servi Get Involved Connect Facebook | Youtube | Flickr | Twitter | Liblogs Join the Liberal Party Have your say on the future of the party Contribute Give what you can and keep the momentum going Volunteer Sign up and help prepare for the next elections Upcoming Events Find out about upcoming events in your area What's New? Ignatieff hopeful of climate change progress, despite Harper snub in Copenhagen Liberals demand Conservatives stop blocking Afghan Special Committee All partners must join together to save Yarmouth
```


Voyant Tools

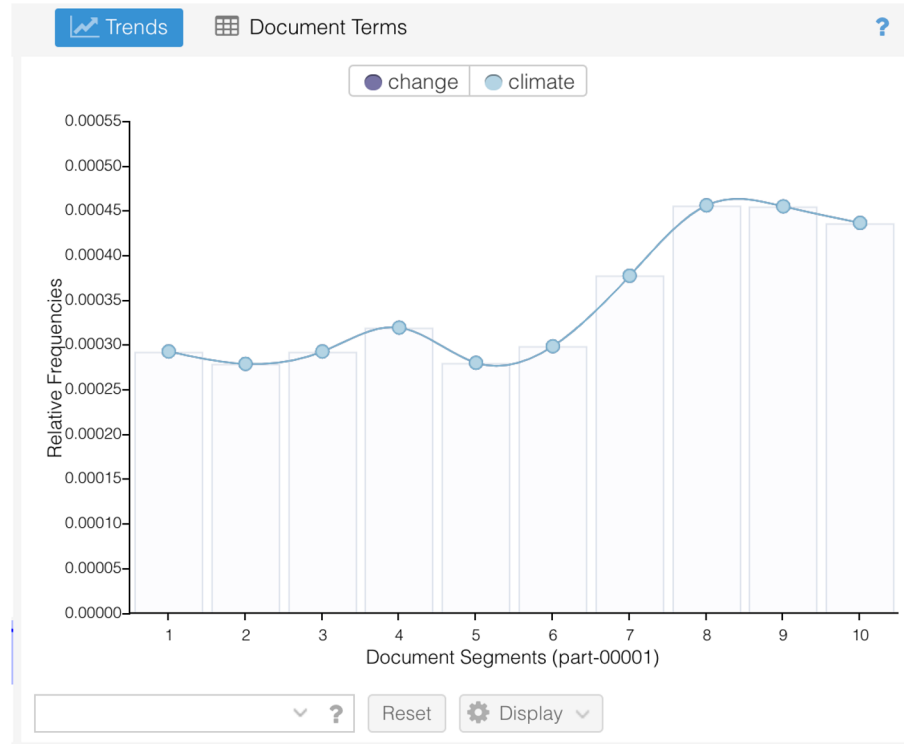


Links

Trends

Contexts

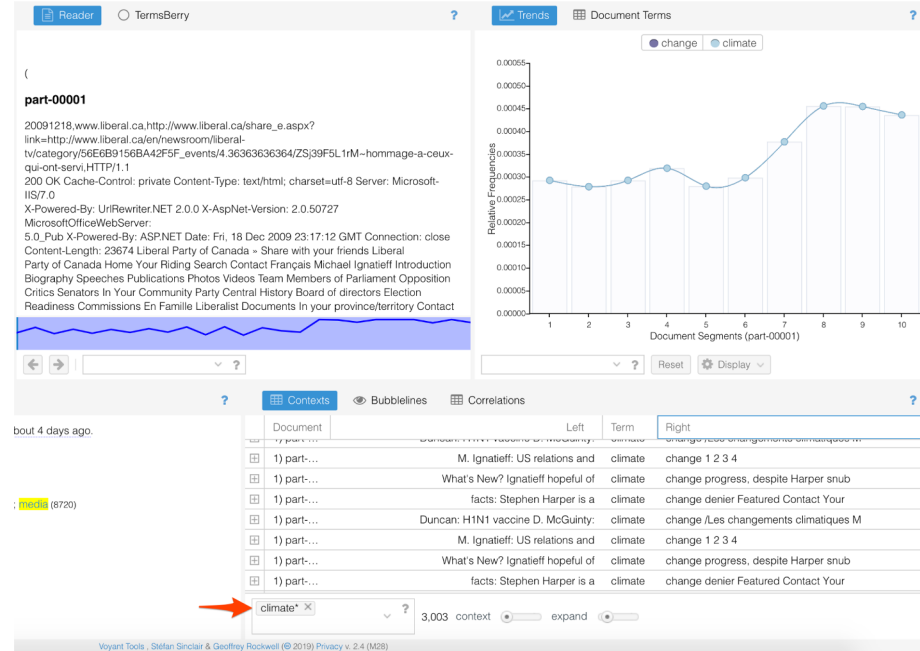
Voyant's Trends tool allows you to graph the frequency of a keyword throughout your text file.



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.



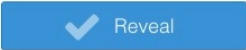



VOYANT

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



[Download](#) [Blog](#) [Wiki](#) [Forum](#) [Support](#) [Bug tracker](#)

[Home](#) [Features](#) [Learn](#) [Develop](#) [Plugins](#) [Services](#) [Consortium](#)

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.

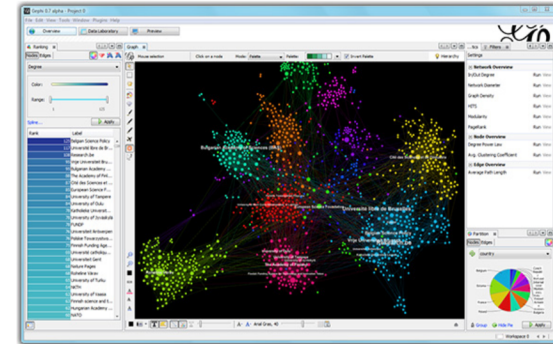
Runs on Windows, Mac OS X and Linux.

[Learn More on Gephi Platform >](#)



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

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



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

Donate



Gephi

Open source visualization and exploration software.

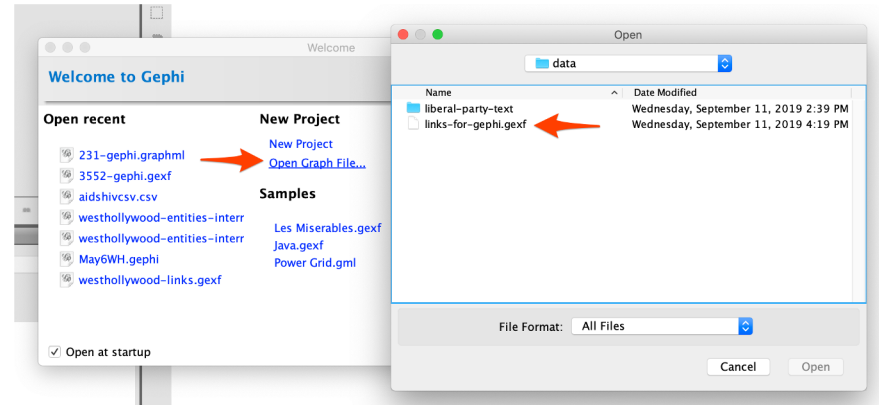
Archives Unleashed Learning Guide:
Network Graphing Archived Websites
with Gephi
<https://cloud.archivesunleashed.org/derivatives/gephi>

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")
```

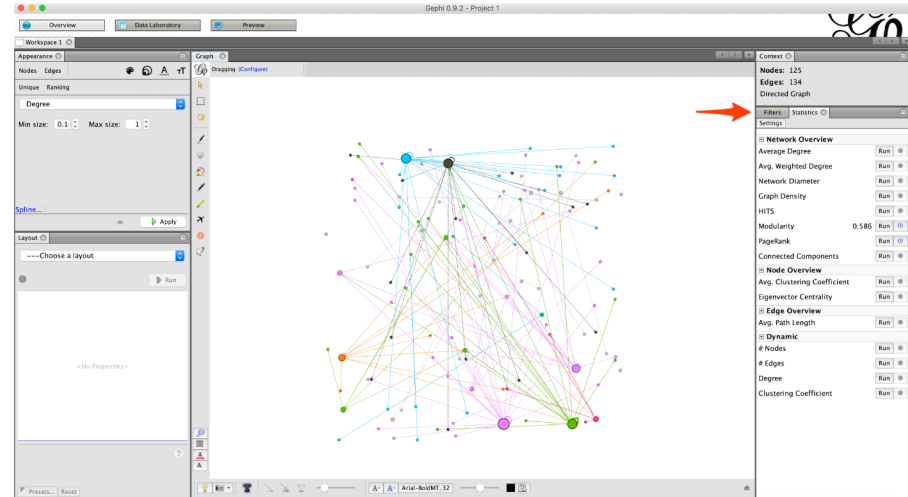


Gephi

Open source visualization and exploration software.

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.





Overview

Data Laboratory

Preview

Appearance

Graph

Nodes Edges

Dragging (Configure)

Context

Nodes:

Edges:

Filters

Statistics

Settings

 Network Overview

Average Degree

Run ●

Avg. Weighted Degree

Run ●

Network Diameter

Run ●

Graph Density

Run ●

HITS

Run ●

Modularity

Run ●

PageRank

Run ●

Connected Components

Run ●

 Node Overview

Avg. Clustering Coefficient

Run ●

Eigenvector Centrality

Run ●

 Edge Overview

Avg. Path Length

Run ●

 Dynamic

Nodes

Run ●

Edges

Run ●

Degree

Run ●

Clustering Coefficient

Run ●

Welcome

Welcome to Gephi



Open recent

- test.gexf
- gephi.csv

New Project

- [New Project](#)
- [Open Graph File...](#)

Samples

- [Les Miserables.gexf](#)
- [Java.gexf](#)
- [Power Grid.gml](#)

 Open at startup

Layout

---Choose a la

<No Properties>

Presets... Reset

A Arial-BoldMT, 32



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>