International Journal of Digital Humanities
https://doi.org/10.1007/s42803-020-00029-6

ORIGINAL PAPER

# From archive to analysis: accessing web archives at scale through a cloud-based interface

Nick Ruest[1] · Samantha Fritz[2] · Ryan Deschamps[2] · Jimmy Lin[3] · Ian Milligan[2]

Received: 14 August 2020 / Accepted: 1 November 2020/Published online: 06 January 2021

**Abstract**
This paper introduces the Archives Unleashed Cloud, a web-based interface for working with web archives at scale. Current access paradigms, largely driven by the scope and scale of web archives, generally involve using the command line and writing code. This access gap means that subject-matter experts, as opposed to developers and programmers, have few options to directly work with web archives beyond the page-by-page paradigm of the Wayback Machine. Drawing on first-hand research and analysis of how scholars use web archives, we present the interface design and underpinning architecture of the Archives Unleashed Cloud. We also discuss the sustainability implications of providing a cloud-based service for researchers to analyze their collections at scale.

**Keywords** Web archives · Interface design · Digital humanities · Accessibility · Sustainability

## 1 Introduction

The importance of web archives for historical research has recently received attention, most notably in two full-length monographs (Brügger 2018; Brügger and Milligan 2018; Hockx-Yu2014; Milligan 2019; Schroeder et al. 2018). Web archives, which consist of web pages and their embedded resources dating back to the mid-1990s that have been collected by organizations such as the Internet Archive and other national libraries, present a profound challenge to historians and other humanities and social sciences researchers who want to study the 1990s or beyond. This is best understood as

✉ Ian Milligan
i2millig@uwaterloo.ca

[1] Digital Scholarship Infrastructure Department, York University, Toronto, ON, Canada

[2] Department of History, University of Waterloo, Waterloo, ON, Canada

[3] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

a combination of opportunities and challenges. Opportunities because we have the potential of more democratic voices included in the historical record: the teenager in the 1990s who wrote a personal home page, corporate webpages from the early 2000s, personal blogs, posts from deployed soldiers, to innumerable other examples. Yet challenge comes in *access*: how can a humanist or social scientist make sense of these resources, which exist on an exponentially different scale than the traditional analog sources that they are used to working with. Consider the challenge that web archives can present: size on the order of petabytes, billions of words, tens of thousands of images, all with murky metadata, provenance, and difficulty to access. Yet it is difficult to imagine a history of the 1990s or beyond where the Web does not feature as a historical source.

Historians are not ready to use web archives, primarily due to these challenges. To use web archives right now essentially means using a Wayback Machine (such as the Internet Archive's implementation at https://archive.org/web/). The Wayback Machine is a replay engine that can be deployed by various collecting institutions, such as national libraries, universities or NGOs and other institutions, to provide access to their web archival collections. A Wayback Machine is great if you know what you are looking for, with its ever-improving keyword search functionality, such as that deployed by the Internet Archive, but it does not scale for more detailed research queries. For example, a researcher may want to do complicated queries (websites that contain certain words and link to certain domains) or exploratory text mining or working with images *en masse*. To do so at present is possible, but requires at the very least the use of command-line interfaces and other techniques associated with the computational humanities. Additionally, processing web archives requires a lot of processing power – but often only during the initial calculations on the raw data, meaning that researchers need surge capacity but only for a few days at most.

So what can be done? In this article, we introduce the Archives Unleashed Cloud as a response to these challenges. The Cloud builds on earlier work with command-line architectures to explore web archives as well as research process cycles to present a web-interface approach to working with web archives at scale (Lin et al. 2017). We argue that by moving towards a cloud-based architecture for web archiving analysis, the challenges of much of web archiving research can be partially surmounted – leaving opportunity. The article does so in three major ways. First, we explore how scholars use web archives and how these usage models do not line up with the existing command-line and developer-focused paradigm of web archive analysis toolkits. Secondly, we explore interface design and technical architecture of the Cloud, with an eye to explaining both how our project works as well as to provide best-practices for other at-scale digital humanities analysis projects. Finally, we address the elephant in the room: sustainability. Ultimately, we argue that the Archives Unleashed Cloud presents both a specific solution to the problems of web archive analysis as well as more general problems of working with data at scale.

## 2 Background and related work

As noted, the current state of access for most users is the Wayback Machine. Within the broader field of web archive access and analysis, however, we are seeing a trend towards accessibility. This can be seen throughout the two main components of the web archiving lifecycle: from *acquisition* to *analysis*. We explore each of these in turn below. Not mentioned at length is that the Wayback Machine itself reflected an early and important move towards accessibility. Launched in 2001, seven years after the Internet Archive's 1996 establishment, the Wayback Machine's ubiquity and simplicity disguises the technical complexity inherent in stitching together images, HTML files, and other resources together in relatively temporally-coherent pages (Ainsworth et al. 2015). Before 2001, users had to use the command line and servers to work with web archives; now we can view them, albeit one by one. In this, we can see that accessibility has long been a part of the web archiving ecosystem.

### 2.1 Capturing web content: from Heritrix to Webrecorder

To be the subject of research, a web archive first needs to be created. The dominant web crawler to create web archival collections is Heritrix, a collaborative open-source project by the Internet Archive and several European national libraries dating back to 2003. Heritrix is free, open-source software that captures web content and embedded objects and saves them in WebARChive (WARC) files, an ISO-standard file format that aggregates all of the resources from a capture (Library of Congress n.d.). Heritrix has a catch, however. While free and open-source, it is difficult to use. A Heritrix user needs to have an advanced level of knowledge of how processing chains work, the various file formats that might be encountered, and most importantly, an understanding of how to debug the odd web behaviour that Heritrix encounters when crawling live content. This generally requires a developer, which is not ideal for researchers or information professionals who might want to preserve web content.

Several services have risen to deal with this complexity. The Internet Archive's Archive-It service provides an easy-to-use curatorial interface that sits on top of Heritrix, providing not only the interface to collect material, but technical support, long-term storage, and ongoing development work which helps deal with the ever-present problem of capturing rapidly-evolving web contents and standards. The downside of Archive-It is that it is a subscription service largely aimed at institutional players, and priced accordingly. It is not meant to be used by individual researchers.

Other services, then, are beginning to appear that let users run their own web crawlers without needing either the institutional support for Archive-It or the developer know-how for Heritrix. Conifer/WebRecorder, available at https://webrecorder.io, allows a user to capture content that they are viewing through their web browser. For example, they could visit http://newyorktimes.com, begin "recording," and capture everything that their browser loads: content, images, videos that are played, and beyond. It is labour intensive, but free and works very well for an individual researcher. Finally, the Internet Archive's "Save Page Now" button also allows a user to request a particular page to be crawled – it is often crawled and made accessible within minutes (previously, users had to make their sites discoverable in the hopes that the Archive would find it), with the caveat that there is currently no way for an end user to access the WARC that is generated with "Save Page Now".

The collection ecosystem then includes several options: the "free" yet hard-to-use-and-deploy option (Heritrix); the supported yet expensive institutional option (Archive-It); and the open-source, easy-to-use yet difficult to scale option (Webrecorder.io). In this we can see both a push towards greater accessibility, as well as being able to meet a wide variety of user needs. The same, unfortunately, is not currently true when it comes to analysis.

## 2.2 Analyzing web content: the archives unleashed toolkit

If we can see good strides towards accessibility in terms of crawling content and then reviewing them (with the Wayback Machine), analysis has largely lagged. One of the main goals of working with web content is to *transform* it into a format usable by digital humanities practitioners. The thought is that most digital humanists will not have heard of a WARC file or have the capacity to work with them at scale. But if text, or hyperlink networks, or entities are extracted from the files, digital humanists and other computational scholars have the capability to work with text or networks at scale using tools such as Python, R, Voyant Tools, or other standard analytical approaches.

Most current web archive analysis projects require detailed knowledge of both the command line as well as how to write code. For example, the code snippet below shows the process by which Emily Kalah Gade, John Wilkerson, and Anne Washington extracted pages with keywords of interest in a corpus of websites using Apache Pig on the now-decommissioned Altiscale research cluster (Gade 2017; Gade et al. 2017).

```
Archive = LOAD "$I_PARSED_DATA" USING SequenceFileLoader()
AS (key:chararray, value:chararray);
Archive = FOREACH Archive GENERATE FROMJSON(value) AS m:[];
Archive = FILTER Archive BY m#`errorMessage' is null;
ExtractedCounts = FOREACH Archive GENERATE m#`url' AS
src:chararray,
SURTURL(m#`url') AS surt:chararray,
REPLACE(m#`digest',`sha1:',”) AS checksum:chararray,
SUBSTRING(m#`date', 0, 8) AS date:chararray,
REPLACE(m#`code', `[^p{Graph]', ` ') AS code:chararray,
REPLACE(m#`title', `[^p{Graph]', ` ') AS title:chararray,
REPLACE(m#`description', `[^p{Graph]', ` ')AS
description:chararray,
REPLACE(m#`content',`[^p{Graph]',` ') AS content:chararray;
UniqueCaptures = FILTER ExtractedCounts BY content MATCHES
`.*naturals+disaster.*' OR content MATCHES `.*naturals+
disaster.*'
OR content MATCHES `.*desertification.*' OR content MATCHES
`.*climates+change.*' OR content MATCHES `.*pollution.*' OR
content MATCHES `.*foods+security.*';
STORE UniqueCaptures INTO `$O_DATA_DIR' USING PigStorage('\
u0001');
```

Gade, Wilkerson, and Washington did amazing work – some of our other work has built on it as a foundation (Wang et al. 2017)– but Apache Pig code is hardly user-friendly. Researchers may learn how to code in this manner, but even if they learn more general-purpose languages like Python or R, Apache Pig might be beyond them.

Confounding things, Pig has largely been eclipsed by Apache Spark – which will in turn be eclipsed in the future by other, newer languages and frameworks. Unlike the relatively stable languages of humanities computing (although the shift from Python 2 to Python 3 is still being felt across the digital humanities), working with data at scale requires using cutting-edge methodologies and approaches. We cannot expect humanists and social scientists to learn a new language every two or three years in order to work with cultural heritage at scale. There are several other projects that aim to provide web archive analytics. ArchiveSpark is an ongoing project that provides an Apache Spark-based analytics framework (Holzmann et al. 2016). Yet they too face the issue of opaque syntax and difficulty to use.

Our own project's Archives Unleashed Toolkit faces this challenge as well. The syntax is slightly easier to read, but is still written in the Scala programming language. For example, this script extracts all of the plain text of a set of web archive files, extracting only those pages crawled in 2008 and 2015:

```
import io.archivesunleashed._
import io.archivesunleashed.udfs._

val dates = Array("2008", "2015")

RecordLoader.loadArchives("/path/to/warcs", sc)
  .webpages()
  .select( "crawldate", extractDomain("url").as("domain"),
  "url", "content")
  .filter(hasDate($"crawl_date", lit(dates)))
  .write.csv("plain-text-date-filtered-2008-2015-df/")
```

While we have made great efforts to make syntax as user friendly as possible, uptake has continued to be very slow. What's wrong with this approach?

On the surface, this approach seemed to meet a demonstrated user need. The Archives Unleashed Toolkit represents a collaboration between computer scientists, historians, and information professionals, who engaged in an iterative co-design process to build an analytics framework that is usable by humanities scholars and social scientists with no formal computer science training (Ruest et al. 2020, Lin et al. 2017). It allows scholars to interrogate web archives in a number of different ways, ranging from crawl statistics to visualizations of web graphs to analyses of frequent mentions of named entities (person names, locations, organizations, etc.). More importantly, it moves beyond keyword search, allowing users to work with data at scale (Jackson et al. 2016). So far, so good.

Yet while successful from a technical perspective, the Archives Unleashed Toolkit has several barriers. Most importantly, it effectively requires knowledge of the command line and previous experience in programming. Scholars need to write or edit Scala scripts to use it. Secondly, even for those conversant with programming and the command line, setting up, configuring, and deploying Apache Spark and the Archives Unleashed Toolkit can be challenging. For scholars with little technical expertise, it is nearly impossible. Even with documentation, we realized that we were still ultimately dealing with complex big data infrastructure that requires a certain level of technical know-how and skill sets that our subject-matter experts, even digital humanists, could not be expected to have.

### 2.3 Uneven pushes towards accessibility

To conclude our review of the field, then, the web archiving environment is characterized by a move towards accessibility and usability of tools, albeit to differing levels based on the stage of the life cycle. When it comes to *collecting* archives, there are a wide variety of approaches that can be adopted: the developer-focused Heritrix approach, the institutionally-focused Archive-It subscription service approach, or light-weight but user-friendly tools such as Webrecorder for individual researcher use. Web archives can be *replayed* using easy-to-use tools such as the Wayback Machine or pywb (as noted above, its ubiquity makes it seem more straightforward than it is).[1] Yet when it comes to *analysis*, options are rather limited. Users are required to open up command line terminals, install software with complicated dependencies, have access to either powerful standing infrastructure or the ability to use cloud services such as Amazon Web Services or Microsoft Azure, if they want to work with web archives at scale beyond replay.

This is unfortunate. Approaches within the digital humanities, broadly defined, have much to offer to the study of web archives. Text analysis scholars could find patterns within these large corpora, from frequently-occurring words or concepts, topics (using strategies such as topic modeling), sentiment analysis, entity extraction, and beyond; network analysis scholars or information retrieval experts might leverage the hyperlink networks to use PageRank or other approaches to find pages of central influence; other scholars might find interest in the composition of the archive itself, such as what was included, what was not, and what this tells us both about the historical moment as well as the collection strategy employed. Indeed, scholars in the digital humanities have shown that you can do much with text, networks, and descriptive metadata (Arnold and Tilton 2015; Graham et al. 2015). In other words, a digital humanist who is versed in computational methods might have a lot of use for large text files, network files in standardized file formats that can be read by network analysis software such as Gephi, or other data contained in open-formats such as JSON, CSV, or HTML. What we cannot expect them to be able to use is data natively stored in the WARC format.

The goal of the Archives Unleashed Cloud is bridging this gap. In other words, how can web archive analysis be accessible in the same way that Archive-It and Conifer/WebRecorder have made crawling accessible without exhaustive technical resources? To do so, we first need to consider how scholars use web archives.

## 3 How do scholars use web archives?

The process model for scholarly interactions developed for the Warcbase project continues to be useful. We call this the Filter-Extract-Aggregate-Visualize(FEAV) cycle (Ruest et al. 2020), an evolution of the Filter-Analyze-Aggregate-Visualize(FAAV) cycle introduced in earlier work (Lin et al., 2017). In short, the FEAV cycle begins with a question from a scholar who wishes to interrogate a web archive. This works as follows:

- Filter: Typically, the scholar begins by focusing on a particular portion of the web archive, selected using both metadata and content. This might be a particular

[1] Pywb is a “Python Web Archiving Toolkit for replay and recording of web archives.” The repository can be found at https://github.com/webrecorder/pywb.

domain, pages with a particular keyword mentioned, or those that link to a particular domain.
- Extract: After finding a subcollection of interest, the scholar then needs to extract some of this information of interest. Examples include extracting links, exploring anchor text, named entities, or all of the plain text.
- Aggregate: Next, the scholar usually wishes to aggregate or summarize the output of the analysis from the previous step. The simplest example of aggregation is counting, e.g., how many times each politician is mentioned, how many links there are from one domain to another, etc.
- Visualize: Finally, the aggregate data are presented in some sort of visualization, which could be as simple as a table of results or as complex as requiring an external application.

We emphasize that this process model is not meant to be prescriptive, but rather offers a reference framework for scholars to get started. Overall, we speak of the FEAV *cycle* because the scholar explores the web archive iteratively through these activities. The Archives Unleashed Cloud is specifically developed to support this process model.

The primary way in which we use the FEAV cycle to work with web archives is through the generation of scholarly derivatives. Digital humanists on the computational side are familiar with working with text, networks, or other standardized file formats – in a way that they are not familiar with WARC files. Through working with humanities and social sciences scholars at a series of eight datathons ("Archives Unleashed" events, held variously in Toronto, Washington DC, London England, San Francisco, and Vancouver), we have found that transforming WARC files into three standardized derivative formats dramatically expands their usability. These are now dealt with  in turn.

The first main file type that scholars want to work with is the plain text of a web archive collection, or the text that appears on all of the HTML pages throughout the web archive. To do so, all of the pages within the collection that are HTML need to be identified. Tags and HTTP headers need to be stripped out of the document (JSoup is the library that our project uses, although BeautifulSoup would be the Python equivalent; header length is stored in the WARC metadata and can be accordingly stripped out using a string operation). A further filter for "200" HTTP response codes is then run, so that 404 pages (the error message that a user receives when they visit an incorrect page) are excluded. Finally, a final step might be the removal of "boilerplate" content: advertisements, navigational bars, and beyond; we do not do this by default, however, as some scholars may be interested in these elements (Kohlschütter et al. 2010).

While this data can be presented in various different outputs, the standard full-text export comes as comma-separated values in the following datafile:

```
crawl_date, domain, URL, MIME type from the server, MIME type
from Apache Tika, language, content
```

For example (hypothetical content):

```
20180115,liberal.ca,https://liberal.ca/en/vote,text/
html,text/plain,en,Ballots are open on December 5th, 2019.
Please vote!
```

```
  20180115,conservative.ca,https://conservative.ca/en/
about-us,text/html,text/plain,en,We are the Conservative
Party of Canada.
```

In the above, the data from each individual record is stored on a single line. While the software supports further filtering, discussed below, these sorts of standard formats allow people to also use their own text analysis, scripting, or programming environment to select data as they see fit. Instead of working directly with the WARC files, they now have data in a familiar CSV format. This can work with most software environments.

The second main file type that scholars want to work with are hyperlink networks. This builds on an understanding that is useful to leverage the structured metadata found within web archives, in this case, the hypertext portion of HTML. Just as the American National Security Agency found it more useful to look at origin and destination for phone calls when allegedly conducting widespread surveillance on their own citizens (Greenwald 2014), when working with web archives links can be seen as votes of confidence or connection between content. We have discussed this at length elsewhere, but in short, one can often learn more about changing link patterns than attempting to read or distantly read extracted text at scale. For example, a page that tends to receive lots of links in a web community might be a community center; a frequently-discussed site; or some other connective tissue (a funding agency, for example, or a news site). While search engine optimization techniques and other forms of spam sites (Google bombing, etc.) are an ever-present concern, various network analysis techniques from PageRank to centrality can help navigate these networks.

This data can similarly be presented in several different outputs, usually taking the rough form of:

```
  date, origin, destination, anchor
```

For example, in the case of the Liberal Party of Canada linking to its opponent, the Conservative Party of Canada:

```
  20180115, liberal.ca/en/our-opponents, conservative.ca/
platform, Conservative Party of Canada Platform
```

A network analysis program can read lists of origin and destination and set up a network analysis. However, we often find that the scale of web archives means that it might be best to aggregate the individual URLs into their domains. In this case, the above example would become:

```
  20180115, liberal.ca, conservative.ca
```

As there are often many recurring domains that link to each other, we also by default do a "count" operation so that the output looks like this:

```
  20180115, liberal.ca, conservative.ca, 39
  20180115, liberal.ca, cnn.com, 10
  20180115, conservative.ca, cnn.com, 5
```

As within network analysis, there are standardized file formats, our software can also natively export to the GEXF or GraphML file formats. These can be opened by programs such as Gephi (https://gephi.org).

The final main file type that scholars want are statistical breakdowns of what have been captured by web archives. This is increasingly important as the provenance of web archives is not documented in a standardized way, so it is very useful to know what has been collected (Maemura et al. 2018). This can help inform an understanding of the first two derivatives. At a minimum, our default outputs look like:

```
domain, number collected
```

i.e.

```
liberal.ca, 30108
conservative.ca, 2005
```

This can help interpret text analysis and give context to what is being found. It can also give a sense of what is *not* present in a web archive.

For convenience, we group the three main derivatives above as “full text,” “network,” and “domain.” Each of them comes in different specific implementations (full text might be separated in different ways, for example by date or by domain), but this is the main typology of what we aim to transform WARC files into. They are all relatively familiar to computational researchers. But, of course, as noted above right now to create all of these file types requires an understanding of the command line, development, and having sufficient infrastructure and processing power to take WARC files and extract these scholarly derivatives. In the next section, we explore the Archives Unleashed Cloud, our platform that enables easy extraction of these derivatives. No more cut-and-pasting of opaque Scala scripts or coding functions – enter our user interface.

## 4 Open-source project and the canonical instance

As seen in Fig. 1, the Archives Unleashed Cloud bridges the gap between easy-to-use curatorial tools like Archive-It and developer-focused analytics toolkits like the Archives Unleashed Toolkit end. It is an open-source project available at https://github.com/archivesunleashed/auk. Development is carried out in the open, with issues, issue templates, pull requests, etc. done in public by both the team and the broader community. Anybody can clone or fork this repository and run their own local or institutional version of the Archives Unleashed Cloud. To run and maintain their own version of the Cloud, however, requires quite a bit of technical expertise and overhead, including access to servers, an understanding of how to deploy a Rails application, and beyond. This would not move most users beyond the complexities of the Toolkit, and hardly meet our goals of accessibility.

Accordingly, much of the emphasis of the Archives Unleashed Cloud project has been running a “canonical instance” in the cloud –https://cloud.archivesunleashed.org. This allows people to use our analytics tools and leverage cloud infrastructure. As noted above, the actual processing and generating of the derivatives is resource-intensive: it takes a lot of computing power to work with the raw WARCs themselves. Once this stage is done, however, analyzing the resulting outputs does not require the same level of computational infrastructure. It thus makes sense for a researcher to turn to an external service when processing WARCs, and then relying on their own laptop or desktop to work with the derivatives. While the Cloud can be deployed in a Hadoop infrastructure, which is what Apache Spark is designed for, we have discovered that in most cases it is quicker and more straightforward to process web archives on a powerful virtual machine. This is because to use Hadoop requires that the files are loaded into the Hadoop Distributed File System, or HDFS. In our practice, a 32-core, 120 GB RAM machine with attached network block storage has been sufficient for processing nearly one petabyte of web archives. Our users are not time insensitive, but if they need to wait for a few days for their collection to be queued up behind others, we feel that is adequate given the batch processing model we employ.

We are thus a free and open-source project with a canonical instance maintained by our team. In some ways, this lets us leverage the best of both worlds: we can harness community input, while still providing physical infrastructure for them to use. That said, it does introduce additional sustainability concerns, discussed below.

## 5 Interface design

Our goal with the Archives Unleashed Cloud, which was collaboratively developed by our interdisciplinary team, was to provide an open-source cloud-based analysis tool that helps researchers and scholars conduct web archive analysis. It supports the priorities of accessibility and usability of web archives by providing users a web-based front end to access and drive the code-heavy Archives Unleashed Toolkit in the back end. The Archives Unleashed Cloud is designed to put the FEAV cycle into action and give researchers access to the scholarly derivatives listed above. In this section, we explore the various views and other interface options that are exposed to users in the Cloud.

Through user engagement, surveys, and testing, we arrived at a basic interface that would use modern web development approaches. The goal was to let people use the core features of working with web archives (generating derivatives and basic analysis) without needing to know domain specific code. We arrived at four main requirements:

- *Syncing metadata about web archive collections* (such as data size, description, public status, URLs, and number of files), which would allow users to see an overview of what they had collected before transferring and running laborious jobs;
- *Transferring data to the interface's back end for analysis*– this would need to be done in a cloud-to-cloud data transfer architecture, as local uploading and downloading of terabytes would be both time consuming and costly;
- *Generating basic scholarly derivatives* such as hyperlinks, full text, and network graphs; and.
- *Allow some in-interface visualization* to assist with research questions and crawl analysis.

We began by whiteboarding out interfaces, and then over a period of months and iterative development, our technical lead and co-investigator designed the site. The final result consisted of a series of "views," giving access to the above core features. Each are discussed in turn.

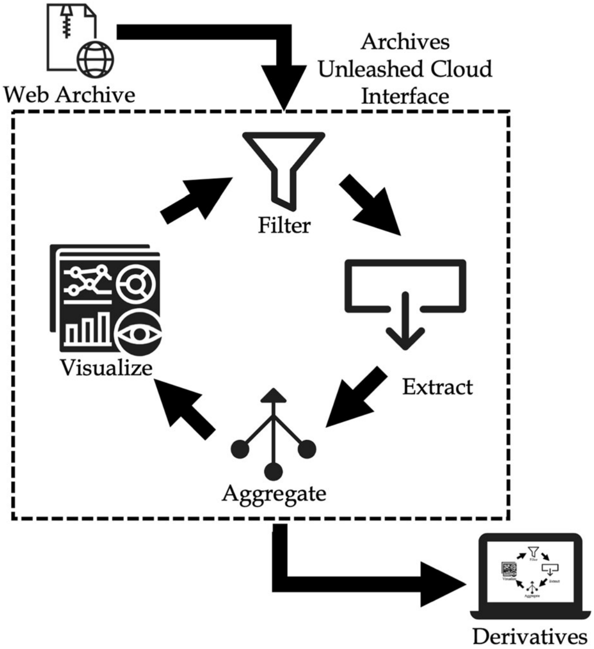


**Fig. 1** The FEAV Cycle in action, with the Cloud as an intermediary between archive and user

### 5.1 Collection overview view

The first requirement was to have a place to highlight metadata from the web archive collections, as noted above. This is important as the size of web archives means that we cannot transfer them over in whole; we need to instead collect information on the basic characteristics of a collection.

When a user arrives at the Archives Unleashed Cloud, after logging in via OAuth services from GitHub or Twitter, they arrive at an initially empty "collections" page (Fig. 2). Users then need to connect the Cloud to their web archive interface by entering credentials and contact information for the Archives Unleashed Cloud (such as institutional information and an e-mail address, useful for both internal project metrics as well as sending notifications via e-mail) as well as for their web archiving service provider.

The Cloud is designed to work with a wide variety of web archiving service providers, and is fundamentally designed around the Web Archiving Systems API, or WASAPI. This Institute of Museum and Library Services-funded API is designed to allow archives to exchange metadata and web archival files between institutions. As of 2020, the only collecting service that supports WASAPI is the Internet Archive's Archive-It subscription service, and Rhizome's Conifer service (formerly WebRecorder). Fortunately, Archive-It ist currently using WASAPI to share WARC files with their clients, suggesting it has become a core function within their organization (which is important for sustainability). While we are in discussions with other providers, including Conifer/WebRecorder, about adding WASAPI interoperability, Archive-It is a natural starting point. A recent National Digital Stewardship Alliance survey found that in 2017, some 94% of surveyed institutions used Archive-It as their main external capture service, a share which has dramatically increased as other providers have closed  (Farrell et al. 2018).

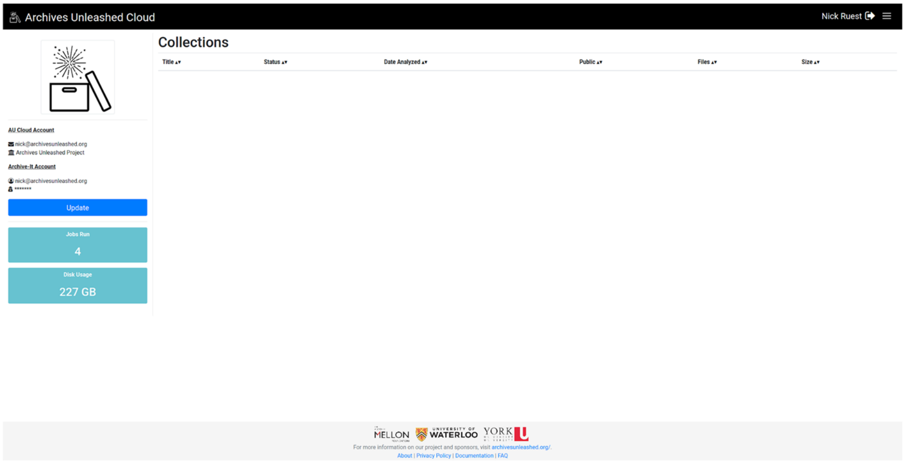
Once a user inputs their credentials, the Cloud interfaces then syncs metadata available in the Archive-It WASAPI files and collections API. Once this sync is complete, it allows the user to analyze individual collections at their convenience.

As this process could take anywhere between a few minutes for a smaller collecting organization to a few hours for a large collecting organization, an e-mail is sent to the user when this job is complete. Once they log in, they can then see all of the information about their web archive collections.

In Fig. 3 we can see the basic "collections" interface once synced with a web archiving service. Account information is at right: we can see a user avatar (via the Gravatar service); information on both the Cloud and Archive-It accounts being used; an option to "update" the metadata should new crawls be run; as well as overall information on activity and disk usage. The majority of the interface as presented on the page is then a table consisting of collection title, the last date that it was analyzed, whether it is public or not (in the Archive-It interface), the number of WARC files, and the size of the collection in megabytes, gigabytes, or terabytes. In some ways, this is the first "filtering" operation that the user does in terms of the FEAV cycle: they now need to select a specific collection to analyze.

### 5.2 Collection analysis view

Before a collection is analyzed, the collection has a blank placeholder with an "analyze collection" button. Selecting "analyze collection" begins a series of operations in the back end, discussed more in detail shortly. In short, the collection is transferred to the Archives Unleashed Cloud for analysis; the Archives Unleashed Toolkit runs a series of operations to extract information in text, hyperlink, and statistical format; and then the raw data of the collection is scheduled for deletion (as the preservation and access copies continue to reside on the Archive-It server).



Collections

| Title ▴▾ | Status ▴▾ | Date Analyzed ▴▾ | Public ▴▾ | Files ▴▾ | Size ▴▾ |
|---|---|---|---|---|---|

AU Cloud Account

nick@archivesunleashed.org
Archives Unleashed Project

Archive-It Account

nick@archivesunleashed.org
*******

For more information on our project and sponsors, visit archivesunleashed.org/.

Fig. 2 An empty collections page

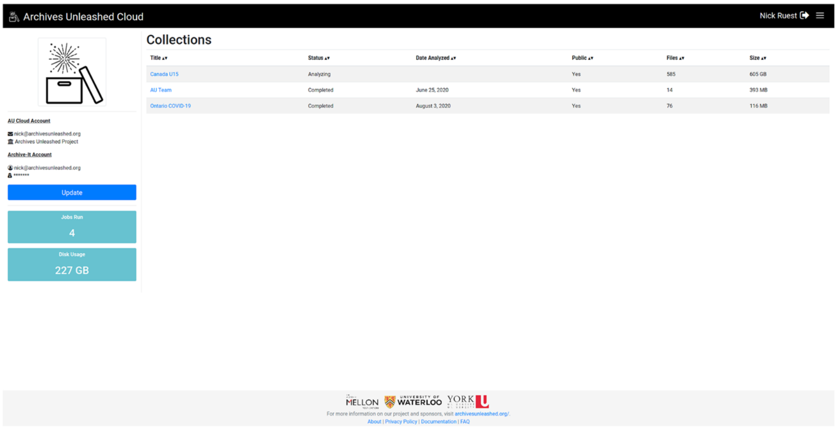
Once the collection is analyzed, users are notified by e-mail, and each collection page is populated with basic information for further analysis. A completed page is seen in Fig. 4.

There are then several options available to the user: derivative downloads, crawl frequency, a network diagram, and basic domain statistics.

Users are initially provided five output files that can be downloaded for further research and analysis: the scholarly derivatives. These include the following as of 2019:

- *Gephi file*. This is a GEXF file, which can be natively loaded into the Gephi network analysis program. GEXF is XML that contains identifiers intended for the layout of the graph and is therefore most suitable for pre-set layouts. While it can be read by others, we highlight Gephi because we have also published tutorials on using web archive data there (discussed below). Due to Graphpass, this file will have basic characteristics already computed and a basic layout.
- *Raw Network file*. This is a GraphML file, which can be loaded into Gephi. Like GEXF, it is also XML-based, but while GEXF was designed for Gephi, graphml was built to have more universal support. As a derivative, we offer GraphML to expand compatibility with third-party software, but no basic layout or calculated characteristics are provided – the user will need to use network analysis programs to explore the file from scratch.
- *Domains file*. This is a straightforward CSV file containing the frequency of domains found within your web archive.
- *Full Text file*. This is (usually) a large text file containing the extracted plain text of all the HTML files found within a collection. We encourage users to consider using open-source approaches like Voyant Tools as a first step when working with these files.
- *Full Text by Domain*. This is a ZIP file containing ten text files corresponding to the plain text of the top ten domains. For example, if the domain "liberal.ca" is the most popular domain in your web archive, there will be a text file called "liberal-ca.txt".

Each of these files is designed to be worked with locally, in the software suite of the user's choice, based on the assumption that it is now in a format that they are familiar with – or can become familiar with using freely-accessible digital humanities resources



Archives Unleashed Cloud

Nick Ruest

Collections

| Title | Status | Date Analyzed | Public | Files | Size |
|---|---|---|---|---|---|
| Canada U15 | Analyzing | | Yes | 585 | 605 GB |
| AU Team | Completed | June 25, 2020 | Yes | 14 | 393 MB |
| Ontario COVID-19 | Completed | August 3, 2020 | Yes | 76 | 116 MB |

AU Cloud Account

nick@archivesunleashed.org
Archives Unleashed Project

Archive-It Account

nick@archivesunleashed.org
*******

Update

Jobs Run
4

Disk Usage
227 GB

MELLON WATERLOO YORK

For more information on our project and sponsors, visit archivesunleashed.org/.
About | Privacy Policy | Documentation | FAQ

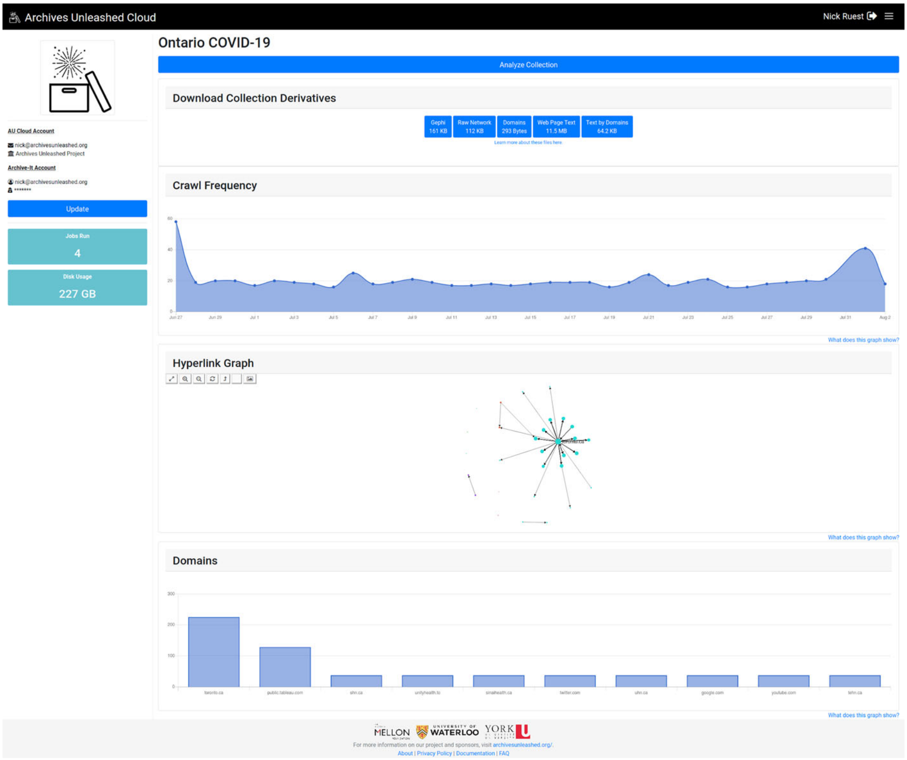
**Fig. 3** A synced collection page

to work with text, networks, or collection summaries. We explore these assumptions in some detail below.

Below the derivatives, users are faced with a crawl frequency chart, which identifies the number of webpages archived for each crawl date within a collection.

The hyperlink diagram is powered by Sigma, a graph drawing JavaScript library, and provides a basic diagram showing how the various crawled domains interact with each other. For more see http://sigmajs.org.

Laying out the diagram proved to be a challenging part of developing this feature for the Archives Unleashed Cloud. In the hyperlink diagram each node (dot) represents a domain (i.e. all of the URLs within a domain such as “torontoist.ca” or “newyorktimes.com”) and each edge (line) represents a link from one node to another. While the infrastructure computed the origin and destination domains as well as the number of times they were linked, actually laying them out in a human-readable format is challenging because the linear algebra required to create metadata that corresponds to the size, color, and position of the nodes is computationally intensive at scale. Software like Gephi begins with a “random layout” or just a cube of nodes and edges, which a user then manipulates to an appropriate diagram. We use our program called GraphPass, available at https://github.com/archivesunleashed/graphpass, to filter networks and provide a default visualization. GraphPass uses the igraph C library to produce visualization-related data in the network files such as color, position, and size based on common social network algorithms (Csárdi and Nepusz 2006).



Archives Unleashed Cloud

Nick Ruest

Ontario COVID-19

AU Cloud Account

nick@archivesunleashed.org
Archives Unleashed Project

Archive-It Account

nick@archivesunleashed.org
********

Jobs Run 4

Disk Usage 227 GB

Download Collection Derivatives

Gephi 161 KB | Raw Network 112 KB | Domains 293 Bytes | Web Page Text 11.5 MB | Text by Domains 64.2 KB

Learn more about these files here

Crawl Frequency

What does this graph show?

Hyperlink Graph

What does this graph show?

Domains

What does this graph show?

For more information on our project and sponsors, visit archivesunleashed.org/.

About | Privacy Policy | Documentation | FAQ

Fig. 4 A basic collections page

Finally, the Cloud offers a chart showing the frequency of the top ten domains present within the collection – essentially, the domains file rendered in the browser. This can help give the user a glance about what they have collected, and can help inform analysis of the full text file or network analysis. It provides additional context.

This relatively straightforward user view, however, belies a lot of the underlying technical complexity. For the user, they are able to use these tools to actualize the FEAV cycle to study their web archives. They can find a collection of interest, analyze it, and then download specific derivatives to work with. Behind the scenes, a complex interplay of data and platforms has combined to make this possible. As of June 2020, 253 users have used the Archives Unleashed Cloud to analyze 912 TB of data over 1,366 collections; testament to the accessible nature of the interface.

## 6 Where do we end and where does the user pick up?

One of the main considerations of the project is where the work of the Archives Unleashed Cloud ends and where the work of the user, or the broader digital humanities or computational humanities community begins. To better explain, let us provide a hypothetical example.

We have a user interested in a particular month of data within a web archive. They download the full text file, which contains the date in yyyymmdd format on each line, and wish to save only the records that have a crawl date of March 15th, 2009 (20090315). We provide information on how to do this using “bash” one liners in the Cloud’s “learning guides” section (typing grep '^(20090315' 1234-fulltext.txt > 20090315-text.txt would create the desired file in either PowerShell or the Mac terminal), as well as link to resources such as the *Programming Historian* which have walkthroughs on doing this. Or should this functionality be baked into the Cloud, requiring additional processing time and power on our end due to our queuing system?

Currently, one workaround is to create pre-generated Jupyter notebooks for users to work with the data – they arrive with code ready to do further processing on the derivative files, breaking them further into dates, days, keywords of interest, and beyond. They are akin to “mad libs,” with our platform filling in some basic information, and a pre-set number of visualizations and text extraction functions are built in. The user simply needs to follow our instructions and change values accordingly. Yet even this requires technical competency. We have resisted turning the Archives Unleashed Cloud into an analysis platform, however, and are instead focusing on the delivery of data to users.

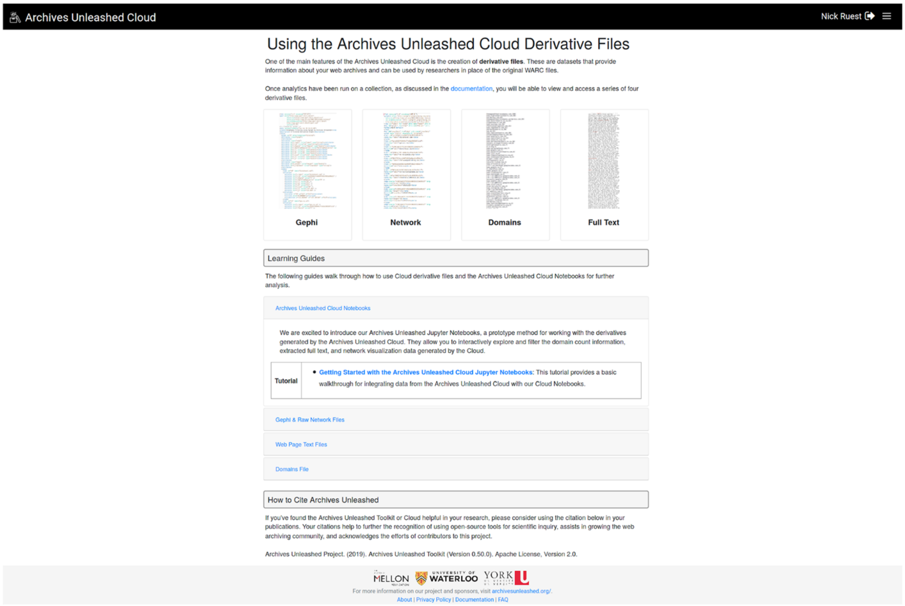
Accordingly, most of our work has focused on pedagogy, curriculum development, and documentation. While the team has run numerous in-person events, both under the banner of the Archives Unleashed project as well as at other conferences and workshops, we are cognizant that most users will not have the funds, interest, or capacity to learn at an in-person event how the Cloud works. We have thus designed our online curriculum as a series of “learning guides,” (see Fig. 5) available prominently on our site, and which are ever-expanding after being tested out online.

As of writing, we have six fleshed-out learning guides that show users how to take the data from our system and then do scholarly research with them. These take two main forms. The first is along the lines of “click here and then click here,” such as introductory notes to using Gephi or filtering full-text files. For users getting started, sometimes a very straightforward and relatively short lesson is what they need to even begin wrapping their head around a concept (this is informed by both in-person teaching as well as one author's years of experience on the core editorial team of the *Programming Historian*). The second are more in-depth reports of using technology to conduct research: using AntConc to explore a collection around the Alberta Oil Sands and finding co-located terms; or using both sentiment analysis or network analysis on a collection of forest fire websites. In this way, we can show how different techniques generate legible historical information. Figure 6 shows one example, where we begin to walk users through different clusters of websites they can find using Gephi, in this case social media and traditional news sites, which were distinct in this collection.

Ultimately, however, the question of how much our platform should do and how much users should do remains unresolved. Where possible, we aim to provide data from our platform in standardized data formats: CSV files for plain text and statistics, or GEXF or GraphML files for network analysis. The project is also open-source and fully public, with bug templates, feature requests, a public and active Slack group, and full-time project staff who can help users with their problems. We always wonder, of course: is this enough?

## 7 Sustainability

The final problem we encounter, like many digital humanities projects, is that of project sustainability (Maron and Loy 2011). Sustainability is an essential process to ensure a project's survival once the (inevitably limited) grant cycle and funding comes to an



Using the Archives Unleashed Cloud Derivative Files

One of the main features of the Archives Unleashed Cloud is the creation of **derivative files**. These are datasets that provide information about your web archives and can be used by researchers in place of the original WARC files.

Once analytics have been run on a collection, as discussed in the documentation, you will be able to view and access a series of four derivative files.

Learning Guides

The following guides walk through how to use Cloud derivative files and the Archives Unleashed Cloud Notebooks for further analysis.

Archives Unleashed Cloud Notebooks

We are excited to introduce our Archives Unleashed Jupyter Notebooks, a prototype method for working with the derivatives generated by the Archives Unleashed Cloud. They allow you to interactively explore and filter the domain count information, extracted full text, and network visualization data generated by the Cloud.

**Tutorial**
- **Getting Started with the Archives Unleashed Cloud Jupyter Notebooks:** This tutorial provides a basic walkthrough for integrating data from the Archives Unleashed Cloud with our Cloud Notebooks.

Gephi & Raw Network Files

Web Page Text Files

Domains File

How to Cite Archives Unleashed

If you've found the Archives Unleashed Toolkit or Cloud helpful in your research, please consider using the citation below in your publications. Your citations help to further the recognition of using open-source tools for scientific inquiry, assists in growing the web archiving community, and acknowledges the efforts of contributors to this project.

Archives Unleashed Project. (2019). Archives Unleashed Toolkit (Version 0.50.0). Apache License, Version 2.0.

For more information on our project and sponsors, visit archivesunleashed.org/.
About | Privacy Policy | Documentation | FAQ

**Fig. 5** Learning guides

end. It is all the more challenging because there is no one model that can cover the wide variety of cases across the community. One essential part of sustainability planning for the Archives Unleashed Project is to understand the financial implications of continuing a service beyond the grant-funded period.

Our sustainability research and plan were accordingly influenced by the context of both the digital humanities as well as open-source tools more generally. We were particularly inspired by the Institute of Museum and Library Services-funded project "It Takes a Village," which explored "Open Source Software Models of Collaboration and Sustainability." This, and other important resources referenced here, gave us sectors to focus on, and an emphasis on the ever-changing and evolving process at play (Arp et al. 2018).

We initially focused on three main avenues to make the project sustainable: defining sustainability, understanding the financial costs at play, and developing a community. We address each of these below, and then briefly note our current long-term strategy.

First, in order to inform sustainability planning, we needed to concretely answer the question "What does sustainability mean to the Archives Unleashed project?" We ultimately defined sustainability to encompass the following three points. First, that the project would remain open-source. Second, that the project would persist as a toolkit and service for web archive analysis. In other words, our goal is to maintain the server and allow people to run analysis on our machines. Thirdly, that we would be able to provide support for structural maintenance and continued, minor development. While major enhancements would require further grant support, we would still need ongoing support to handle security warnings, package deprecations, updates, server maintenance, and the like. While answering this question it was important to consider the various areas of sustainability, economical (cost of sustaining the project), social (community support and involvement), cultural (widely held practices and process),

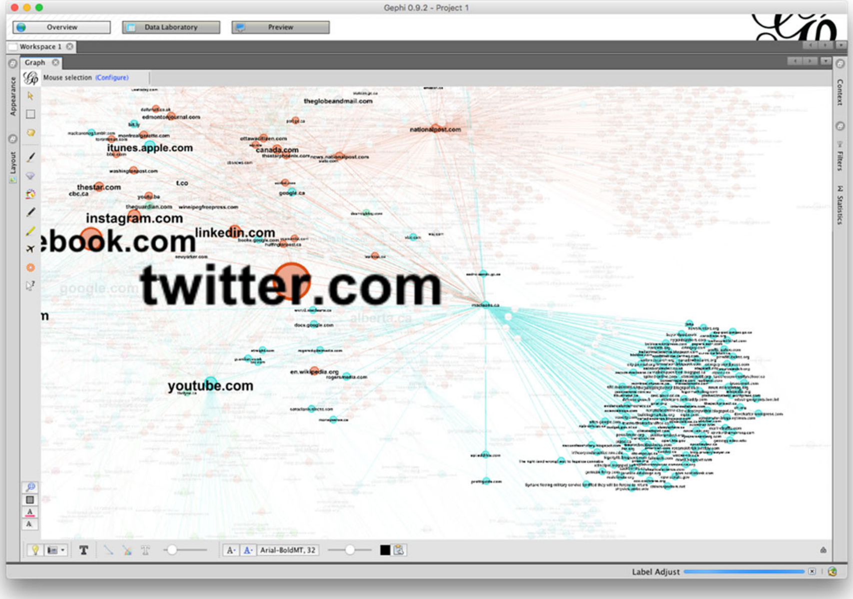


Fig. 6 A screenshot from our Gephi learning guide, showing clusters in the Fort McMurray collection

and legal (licensing and legal risk management). Our planning has focused on the first two and is discussed below (Helander and Antikainen 2006).

The second main sustainability activity was to conduct a cost analysis to understand the real financial costs associated with the transfer, storage, and analysis of web archives within a cloud-based infrastructure (Deschamps et al. 2019). From the Cloud's inception we collected performance logs, detailing the time intervals for generating derivatives and overall processing time, allowing us to calculate the cost to run all operations independently of university infrastructure with Amazon Web Services. From this, we learned that our rough cost of US$7 per terabyte of web archive analyzed (whether it is to analyze a web archive to create plain text derivatives, network diagrams, or beyond) is our bottom-line figure for financial sustainability. In other words, to ensure the ongoing operation of the Cloud, at minimum we would need to find a cost recovery model that supported us at this level with some additional margin for overhead. There are several revenue models that have been used successfully within the digital humanities, open-source, and open academic resource environment, including the community model, subscription model (such as memberships and pay-per-use), commercial model, and central support models (Chang et al. 2007; Guthrie et al. 2008). When it comes to deciding on a model, the goal will be to implement one that matches the financial support needed to maintain the project, but also one that allows it to continue to grow and thrive.

Thirdly, we realized that sustainability efforts are meaningless without a community base, which for our project includes users, contributors, and maintainers. Community development and sustainability go hand-in-hand; participants assist in improving functionality, and mature software can attract a larger user/contributor/developer base (Nyman and Lindman 2013). Crucial to the project's success is leveraging an open-source ecosystem, and to that end, regional datathons staged under the Archives Unleashed banner have been vital to ensure broad community buy-in and continued involvement. The datathon model brings together researchers, programmers, visualization experts, graphic designers, and others into one room in order to facilitate their intensive collaboration on a shared project. In our case, programmers, academics, memory institution professionals, and other librarians gather to work on accessing web archives with our Cloud interface. For sustainability purposes, the datathons generate awareness of the platform, provide opportunities for the community to learn how to use the Cloud, explore new research possibilities, and nurture continued engagement with the platform and broader community. To date, we have run three Toolkit and Cloud-focused events in Toronto, Vancouver, and Washington DC (following on from four earlier events), and each has seen an influx of new users, suggestions, and engagement with the project.

As our project is still in the final stage of grant funding, we do not have any magic bullets to offer in this article. As this article approached the submission period, the Archives Unleashed project was delighted to announce our next stage of the project: partnering with the Internet Archive's Archive-It service to provide a long-term institutional home. We believe this fits in well with our project's overall approach to sustainability: a realization that sustainability is a complex suite of multifaceted layers, and thoughtful consideration of several factors will impact a project's success, from governance, documentation, and marketing, to collaboration, finance, legal and more technical elements such as code quality, support, security, and dependency hygiene (Nesbitt 2017). However, by discovering the basic costs needed to cover resource support, exploring methods of building an active community, and defining what success looks like, we have articulated the beginnings of an economic and context analysis. The final question, as always, is whether these costs are palatable to institutions or researchers. Nothing can ever be free (Guthrie et al. 2008).

## 8 Conclusion

While librarians, archivists, and other curators have been rapidly collecting data – using a variety of services across the web archive collection ecosystem – research access has lagged. Simply put, scholars do not have the tools needed to facilitate the kind of access that they require. Once a scholar wants to go beyond the URL and keyword-focusedone-page-at-a-time interface of a Wayback Machine, if they do not have advanced computing skills and access to infrastructure, they have few to none options available to them.

This project, the Archives Unleashed Cloud, unlocks the potential of web archives by developing and providing the tools, via a cloud service, for scholars with limited – but not none – technical expertise to explore archived web content. In this article, we have introduced the Cloud, as well as our particular educational, sustainability, and scoping challenges. Our hope is that not only we will advance the conversation amongst web archive practitioners, but also digital humanists who need to use cloud platforms to advance their inquiries into a wide variety of fields.

**Funding** This research was primarily supported by The Andrew W. Mellon Foundation [grant number 21700701]. Additional support was from Compute Canada, Start Smart Labs, York University, and the University of Waterloo.

## References

Ainsworth, S. G., Nelson, M. L., & Van de Sompel, H. (2015). Only One Out of Five Archived Web Pages Existed As Presented, *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, Guzelyurt, Northern Cyprus, September 2015.

Arnold, T., & Tilton, L. (2015). *Humanities data in R: exploring networks, geospatial data, images, and text.* New York: Springer.

Arp, L. G., Forbes, M., Cartolano, R. T., Cramer, T., Kimpton, M., Skinner, K., & Whiteside, A. B. (2018). It Takes a Village: Open Source Software Sustainability. Report. *LYRASIS*. https://www.lyrasis.org/programs/Documents/ITAV_Interactive_Guidebook.pdf. Accessed 17 Nov 2020.

Brügger, N. (2018). *The archived web: doing web history in the digital age*. Cambridge: MIT Press.

Brügger, N., & Milligan, I. (Eds.). (2018). *SAGE handbook of web history*. London: SAGE.

Chang, V., Mills, H., & Newhouse, S. (2007). From Open Source to long-term sustainability: Review of Business Models and Case studies. In Chang, V. (ed.), *Proceedings of the UK E-Science All Hands Meeting 2007*. University of Edinburgh/University of Glasgow (acting through the NeSC).

Csárdi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems,1695*(5), 1–9.

Deschamps, R., Fritz, S., Lin, J., Milligan, I., & Ruest, N. (2019) The Cost of a WARC: Analyzing Web Archives in the Cloud, *Proceedings of the 19th ACM/IEEE-CS on Joint Conference on Digital Libraries*. Champaign, IL.

Farrell, M., McCain, E., Praetzellis, M., Thomas, G., & Walker, P. (2018). Results of a Survey of Organizations Preserving Web Content. Report. *National Digital Stewardship Alliance*. https://osf.io/ht6ay/. Accessed 17 Nov 2020.

Gade E (2017) Replication data and code for "The.GOV Internet Archive: A Big Data Resourcehttps://doi.org/10.7910/DVN/YINHYLGade, E. (2017). Replication data and code for "The.GOV internet archive: a big data resource for political science." https://doi.org/10.7910/DVN/YINHYL.

Gade, E. K., Wilkerson, J., & Washington, A. (2017). The.GOV internet archive: a big data resource for political science, *Political Methodologist*. https://thepoliticalmethodologist.com/2017/03/16/the-gov-internet-archive-a-big-data-resource-for-political-science/.Accessed 17 Feb 2019.

Graham, S., Milligan, I., & Weingart, S. (2015). *Exploring big historical data: the historian's macroscope*. London: Imperial College Press.

Greenwald, G. (2014). *No place to hide: Edward Snowden, the NSA, and the U.S. Surveillance State*. New York: Metropolitan Books.

Guthrie, K., Griffiths, R., & Maron, N. (2008). *Sustainability and revenue models for online academic resources*. Report. Ithaka. https://sr.ithaka.org/wp-content/uploads/2015/08/4.15.1.pdf. Accessed 17 Nov 2020

Helander, N., & Antikainen, M. (2006). *Essays on OSS practices and sustainability (No. 36)*. Tampere: eBRC Research Reports.

Hockx-Yu, H. (2014). Access and scholarly use of web archives. *Alexandria,25,* 113–127.

Holzmann, H., Goel, V., & Anand, A. (2016). ArchiveSpark: Efficient Web Archive Access, Extraction and Derivation, *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, Newark, NJ, June 2016.

Jackson, A., Lin, J., Milligan, I., & Ruest, N. (2016). Desiderata for Exploratory Search Interfaces to Web Archives in Support of Scholarly Activities, *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, Newark, NJ, June 2016.

Kohlschütter, C., Fankhauser, P., & Nejdl, W. (2010). Boilerplate detection using shallow text features, *Proceedings of the Third ACM International Conference on Web Search and Data Mining - WSDM '10*, New York, NY.

Library of Congress. (n.d.). WARC, Web ARChive file format. DigitalPreservation.Gov. http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml. Accessed 20 Feb 2019.

Lin, J., Milligan, I., Wiebe, J., & Zhou, A. (2017). Warcbase: scalable analytics infrastructure for exploringweb archives. *ACM Journal of Computing and Cultural Heritage, 10*(4), Article, 22*,* 1–30.

Maemura, E., Worby, N., Milligan, I., & Becker, C. (2018). If these crawls could talk: Studying and documenting web archives provenance. *Journal of the Association for Information Science and Technology, 69*, 1223–1233.

Maron, N. L., & Loy, M. (2011). Funding for sustainability: how funders' practices influence the future of digital resources. Report. *JISC Strategic Content Alliance*. https://sca.jiscinvolve.org/wp/files/2011/06/examination_funder_polices_practices_UK.pdf. Accessed 17 Nov 2020.

Milligan, I. (2019). *History in the age of abundance? How the web is transforming historical research*. Kingston: McGill-Queen's University Press.

Nesbitt, A. (2017). What does a sustainable open source project look like? Libraries.io. https://medium.com/libraries-io/what-does-a-sustainable-open-source-project-look-like-bf9b8cf824f8. Accessed 21 Feb 2019.

Nyman, L., & Lindman, J. (2013). Code forking, governance, and sustainability in open source software, technology. *Innovation Management Review,* (2013), 7-12.

Ruest, N., Lin, J., Milligan, I., & Fritz, S. (2020) The Archives Unleashed Project: Technology, Process, and Community to Improve Scholarly Access to Web Archives, *Proceedings of the 20th ACM/IEEE-CS on Joint Conference on Digital Libraries*, Wuhan, China, August 2020.

Schroeder, R., Brügger, N., & Cowls, J. (2018). Historical web as a tool for analyzing social change. In J. Hunsinger, L. Klastrup, & M. M. Allen (Eds.), *Second international handbook of internet research* (pp. 1–16). Dordrecht: Springer Netherlands.

Wang, Z., Lin, B., Milligan, I., & Lin, J. (2017). Topic shifts between two US Presidential Administrations. Unpublished paper on research blog. https://ianmilligan.ca/2017/07/04/topic-shifts-between-two-us-presidential-administrations/. Accessed 21 Feb 2019.