# Web Archive Profiling Via Sampling
# Final Report

Sawood Alam[1], Michael L. Nelson[1], Herbert Van de Sompel[2], Lyudmila L. Balakireva[2], Harihar Shankar[2], Nicolas J. Bornand[2], and David S. H. Rosenthal[3]

[1]Computer Science Department, Old Dominion University, Norfolk, VA (USA)
[2]Los Alamos National Laboratory, Los Alamos, NM (USA)
[3]Stanford University Libraries, Stanford, CA (USA)

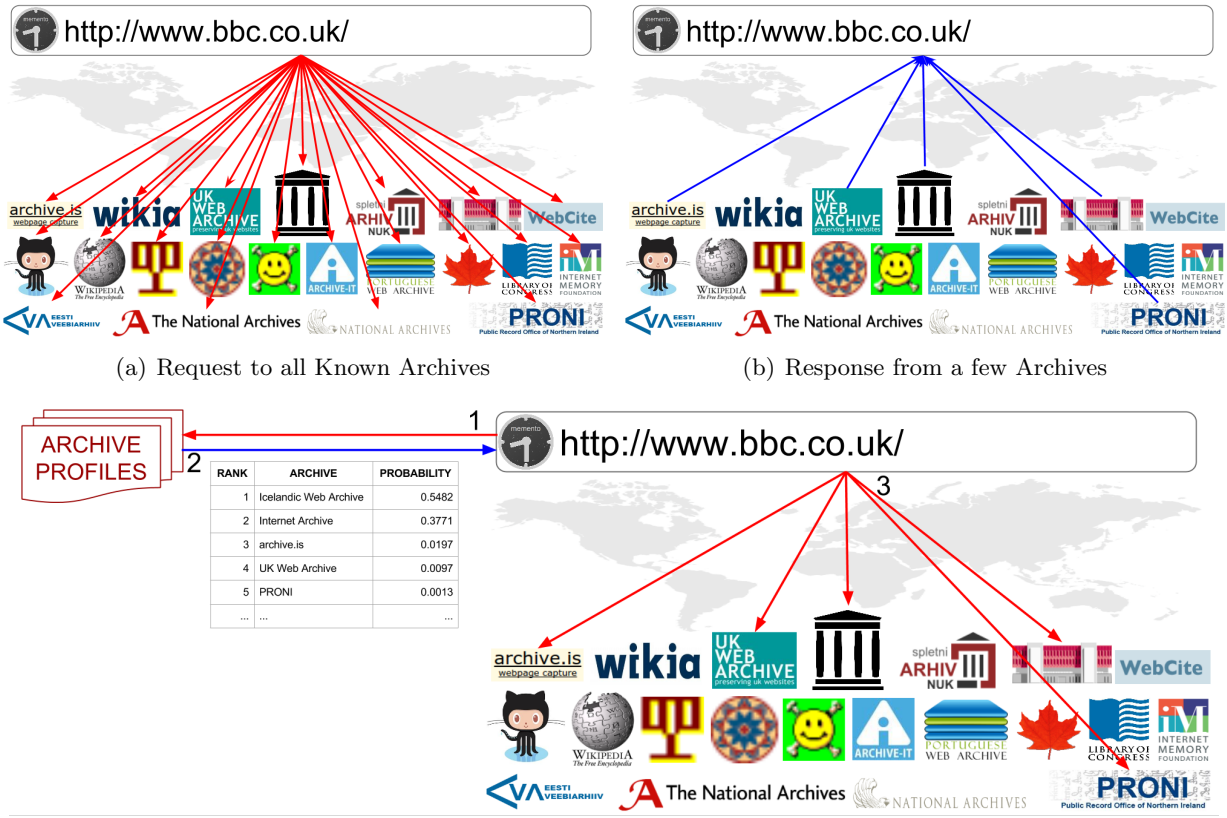
September 16, 2016

## 1 Summary

This report covers the results, deliverables, and ongoing status of the International Internet Preservation Consortium (IIPC)-funded project "Web Archive Profiling Via Sampling"[1], a joint project between Old Dominion University (ODU), Los Alamos National Laboratory (LANL), and Stanford University, and 18 month project which ran from September 2014 – March 2016. Although the project has ended, the research directions uncovered in the course of this project continue, most notably in the ongoing PhD research of Sawood Alam. This report summarizes the contributions and deliverables of the project, with links to code, datasets, presentations, and papers as appropriate. Some of the highlights of the project include:

- A light-weight, machine learning approach to profiling web archives based only on their responses to past queries. This approach was presented at JCDL 2016 [13] and put into production at LANL[2].
- The open source MemGator Memento Aggregator, also presented at JCDL 2016 [4]. This Memento Aggregator code base is written in Go and is designed for rapid deployment of a Memento Aggregator suitable for customization and integration into local projects. Additionally, it is distributed as pre-compiled cross-platform portable binaries. MemGator is being used in various open source projects, including NetCapsule and WAIL.
- The CDXJ Format [11] for serializing the common archiving CDX format in newline-separated json. CDXJ is seeing widespread adoption in the community, including the popular Python WayBack (PyWB) and actively being pushed for the upcoming OpenWayback 3 software[3].

[1] `http://www.netpreserve.org/projects/web-archive-profiling-sampling`
[2] `http://timetravel.mementoweb.org`
[3] `https://iipc.github.io/warc-specifications/specifications/cdx-format/openwayback-cdxj/`




(a) Request to all Known Archives

(b) Response from a few Archives

(c) Memento Aggregator with Profile Based Routing

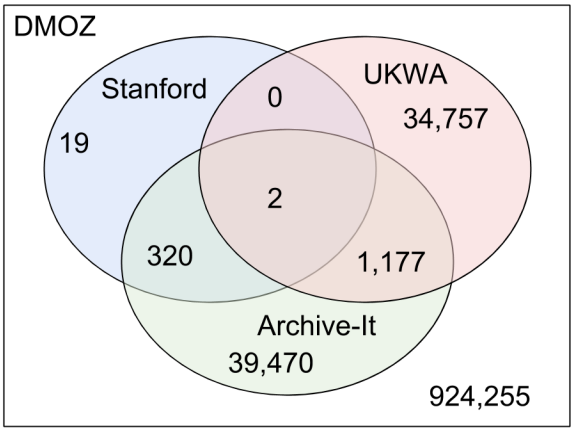
Figure 1: Request-Response Cycle of Memento Aggregator and Various Archives

- Since evaluating archival summaries is highly dependent on the sample inputs, we have collected various large-scale URI datasets from the open web and IIPC members that can be used to standardize evaluation of profiling strategies.

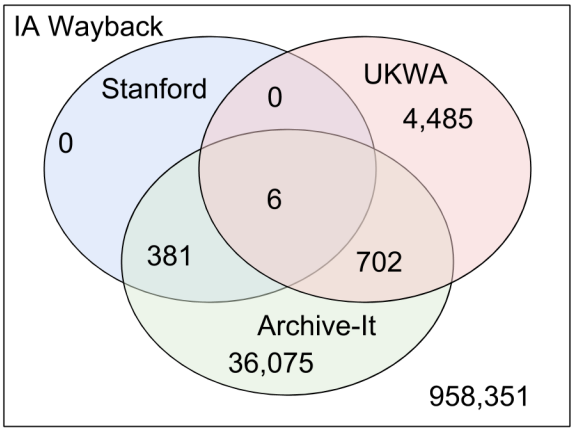
## 2 Overview

The number of public web archives supporting the Memento protocol[4] natively or through proxies continues to grow. Currently, there are 28 publicly available web archives[5], with more scheduled to support Memento in the near future. Figures 1(a) and 1(b) illustrate a naive implementation of the Memento Aggregator where each request is broadcasted to all the known archives, but only a few archives return good results. The Memento Aggregator, the Time Travel Service[6], and other services, both research and production, need to know which archives to poll when a request for an archived version of a file is received. An efficient Memento routing in aggregators is desired from both aggregators' and archives' perspective. Aggregators can reduce the average response time,

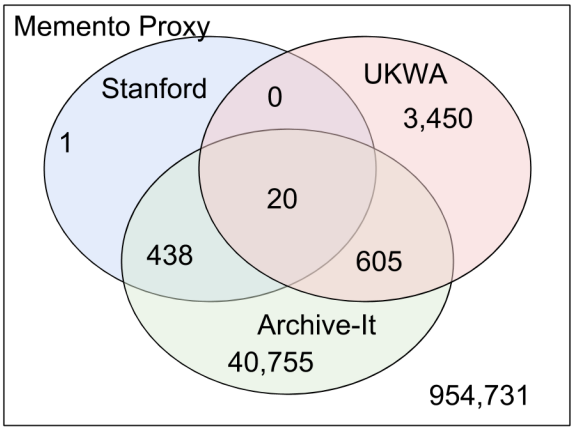
[4] https://tools.ietf.org/html/rfc7089
[5] http://labs.mementoweb.org/aggregator_config/archivelist.xml
[6] http://timetravel.mementoweb.org/



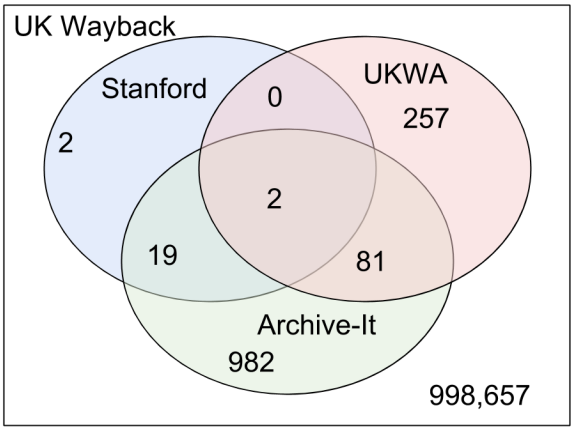

(a) Overlap of DMOZ.




(b) Overlap of IA Wayback.




(c) Overlap of MementoProxy.




(d) Overlap of UK Wayback.

Figure 2: Sample Query URI-R Overlap in Various Archives (1M URIs in each sample)

improve overall throughput, and save network bandwidth. Archives benefit by the reduced number of requests for which they have no holdings, hence saving computing resources and bandwidth.

For example, in December 2015 with the introduction of the emulation service OldWeb.today[7], many archives struggled with the increased traffic. We found that fewer than 5% of the queried URIs are present in any individual archive other than the Internet Archive as illustrated in Figure 2. In this case, being able to identify a subset of archives that might return good results for a particular request becomes very important.

Figure 1(c) illustrates a setup where a Memento Aggregator queries an archive profile service and gets a list of archives sorted in the order of the probability of finding a copy of the queried file in them so that the aggregator can choose the top-K archives from the list to make the requests. Previous work proved that simple rules are insufficient to accurately model a web archive's holdings.

[7] `http://oldweb.today/`

For example, simply routing requests for `*.uk` URIs to the UK National Archives is insufficient: many other archives hold `*.uk` URIs, and the UK National Archives holds much more than just `*.uk` URIs. This is true for the many other national web archives as well.

In this work we researched a variety of methods for determining the holdings of an archive by sampling its contents. In particular, we will look at how archives respond to queries for archived content and over time build up a profile of the top-level domains (TLDs), Uniform Resource Identifiers (URIs), content language, and temporal spread of the archive's holdings. We provided less expensive and less intrusive ways for generating archive profiles that converge over time to that generated by directly measuring an archive's contents, defined a format and mechanism for archival profile serialization and dissemination. The associated techniques and formats are suitable for both IIPC and non-IIPC archives. We developed (as open-source) various scripts and tools to generate profiles, analyze profiles, and aggregate various archives efficiently.

# 3 Deliverables

Following is an itemized list of deliverables as described in the original proposal. Throughout the project period, we maintained a project progress report page[8]. Additionally, we made various documents available to IIPC members for review as we worked on various tasks outlined in the proposal.

## 3.1 D1: baseline development, testing with IIPC members

- defined various profiling policies
- defined metrics, structure, and terminology for profiles
- implemented various ways to generate profiles from CDX
- a configuration mechanism to describe the archive
- wrote data cleanup code
- wrote various analysis code
- a simple entry point code for archives to generate profiles
- Ahmed AlSum ran the code and generated various profiles for Stanford archive

We presented the initial work at IIPC GA 2015 [6, 1], and a full paper in TPDL 2015 [15] which was extended and published in IJDL [16]. This initial work was also presented or published in various other places [7, 2, 3, 14, 8] and later will be presented in [9, 10]. All the related code is publicly published[9].

[8] `http://www.cs.odu.edu/~salam/presentations/iipc/progress.html`
[9] `https://github.com/oduwsdl/archive_profiler`

### 3.2 D2: sample URI collection, dissemination and feedback from IIPC

- collected four million URIs from DMOZ, IA Wayback logs, ODU Memento Proxy logs, and UKWA access logs
- scraped Reddit posts data and extracted one million URIs
- generated profiles using fulltext search for top keywords and random sampling
- implemented a language and collection agnostic random searcher for sampling URIs and profiling
- generating profiles via URI sampling where no fulltext search is available

We plan to publish the extracted sample URIs from various sources (after some cleanup)[10]. The Reddit scraper code is made publicly available[11]. Our full-text search based profiling work was presented at TPDL 2016 [17].

### 3.3 D3: collecting/sampling query logs from IIPC members

- acquired anonymized IA Wayback access logs of one year
- acquired LANL Memento aggregator query logs of 2+ years
- have ODU Memento aggregator query logs of many years
- acquired anonymized UKWA access log sample
- formally asked IIPC members for their access/query logs via email and a blog post
- acquired access logs from OldWeb.today, UK National Archives, and Stanford University Archive

We plan to publish the extracted sample URIs from various access logs (after some cleanup). To collect these logs we sent emails to the IIPC members mailing list and published a blog post[12] which generated a good response such as DSHR posted a commentary on his blog[13] while Nicholas Taylor (Stanford University Libraries), Graham Seaman (UK National Archive), and Ilya Kreymer (Rhizome) offered access logs.

### 3.4 D3: instrumenting Memento aggregator

- an initial code to consume profiles and return ordered list of archives
- implemented a brand new Memento aggregator called MemGator
- implemented feature in the MemGator to utilize the ordered list of archive

[10] `https://github.com/oduwsdl/SampleURLs`

[11] `https://github.com/ibnesayeed/reddit-scraper`

[12] `https://netpreserveblog.wordpress.com/2016/01/08/memento-help-us-route-uri-lookups-to-the-right-archives/`

[13] `http://blog.dshr.org/2016/01/aggregating-web-archives.html`

- implemented various binary classifiers to route Memento queries

MemGator code and binaries are made available publicly[14] along with the Docker image[15]. We have also created a browser based MemGator terminal for demonstration purposes[16] and the corresponding Docker image[17]. MemGator demo and poster was presented at JCDL 2016 [12, 13] and briefly introduced at the two "Archives Unleashed" Hackathons [5, 4]. The binary classifier related paper (described further below) was also presented at JCDL 2016 [18].

### 3.5 D3: other dimensions for profiling

- basic work on time profiles
- performed analysis of suitable sample size
- basic work on language profiles

Relevant code and data is present in the archive profiler repository. The analysis will be published in upcoming publications, including Sawood's PhD dissertation.

### 3.6 D3: internal crawler

- discussed possibilities to implement this
- discussed alternate approaches to surface dark archive holdings
- The CDX profiler is generalized so it can even work on a list of URIs that can be used to generate profiles for dark and private archives

The archive profiler repository contains the relevant code to generate profiles from various sources.

### 3.7 D3: analysis, simulation, validation

- performed resource requirement analysis
- performed growth analysis
- performed cost and precision analysis
- validated effect of various profiling policies in predicting presence of Mementos in archives
- analyzed precision, specificity, accuracy, and recall tradeoff

Detailed analysis of repository growth and cost was published in TPDL, WADL, and JCDL [15, 16, 17], and will continue to be included in upcoming publications. Various analysis scripts are included in the archive profiler repository along with some sample data.

[14] `https://github.com/oduwsdl/memgator`
[15] `https://hub.docker.com/r/ibnesayeed/memgator/`
[16] `https://github.com/ibnesayeed/MemGatorTTY`
[17] `https://hub.docker.com/r/ibnesayeed/memgatortty/`

### 3.8 D4: serialization, transfer, collecting IIPC feedback

- implemented JSON-LD serialization, but discarded due to scale related issues
- defined CDXJ format for serialization
- generated 23 different profiles for each of the two archives and three sample query sets
- implemented a way to push profiles in a GitHub repository automatically
- verified file size limits in GitHub and other places
- profile storage and dissemination options discussed
- formally introduced the CDXJ and ORS serialization formats
- a GitHub fork based workflow is implemented for profile dissemination and discovery

The CDXJ format was first presented and discussed in IIPC GA 2015 [1], then an introductory blog post was published as a motivation to formalize the format and collect feedback[18], and finally we started working on a format specification draft [11]. GitHub fork based dissemination and discovery mechanism is programmed in the main archive profiler code.

## 4 Outcomes from the Original Proposal Tasks

This section summarizes the various tools, scripts, and datasets that were produced in the execution of tasks D1–D4.

### 4.1 Archive Profiler

Archive profiler[19] contains scripts to generate profiles of any archive with different profiling policies from different sources such as CDX files, list of archived URIs, access logs, full-text search, or a URI sample. It also contains some scripts to analyze profiles. Figure 3 illustrates a sample profile generated using the archive profiler script. We thank Ahmed AlSum (Stanford University Archive) for testing the code on Stanford collections.

### 4.2 Eight URI Samples

Eight URI samples[20], containing one million URIs each, were extracted from extracted from eight different sources including: DMOZ, Reddit, and six different archive/aggregator server logs. We thank Andy Jackson (British Library), Nicholas Taylor (Stanford University Libraries), Graham Seaman (UK National Archive), and Ilya Kreymer (Rhizome) for providing useful access logs.

[18] `http://ws-dl.blogspot.com/2015/09/2015-09-10-cdxj-object-resource-stream.html`
[19] `https://github.com/oduwsdl/archive_profiler`
[20] `https://github.com/oduwsdl/SampleURLs`

```
@context ["https://oduwsdl.github.io/contexts/archiveprofile"]
@id {"uri": "http://www.webarchive.org.uk/ukwa/"}
@keys ["surt_uri"]
@meta {"name": "UKWA 1996 Collection", "type": "urikey#H3P1", "...": "..."}
com,dilos,)/region {"frequency": 14, "spread": 2}
edu,orst,)/groups {"frequency": 3, "spread": 1}
uk,ac,rpms,)/ {"frequency": 124, "spread": 1}
uk,co,bbc,)/images {"frequency": 152, "spread": 3}
```

Figure 3: Sample Profile in CDXJ Format

### 4.3 Internal Crawler

The original proposal deliverables included an internal crawler to profile dark archives. For practical reasons, we realized that a crawler that explores the collections of dark archives and builds profiles would cause difficulties such as scoping the crawl, running on premise, and slow discovery. So, we addressed the problem differently and generalized our archive profiler script to build profiles not just from CDX files, but from access logs and plain lists of archived URIs as well. This approach simplifies the task and does not require any special expertise or system requirement that a custom crawler might need.

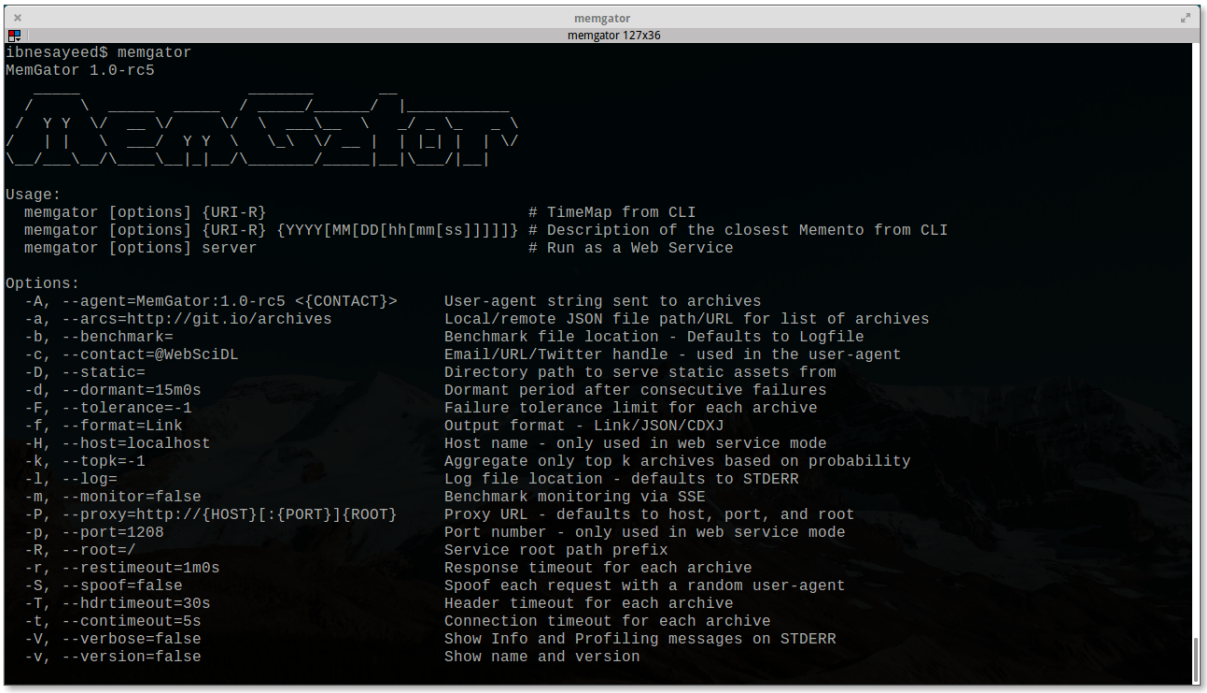
## 5 Additional Outcomes Not in the Original Proposal

As we worked on the project and gained insight, opportunities presented themselves based on the lessons we learned during the course of the project. The following is a list of contributions to the web archiving community that do not correspond our original research plan; they represent deliverables or strategies above and beyond what we initially anticipated being able to deliver and thus do not correspond to a section in the original proposal.

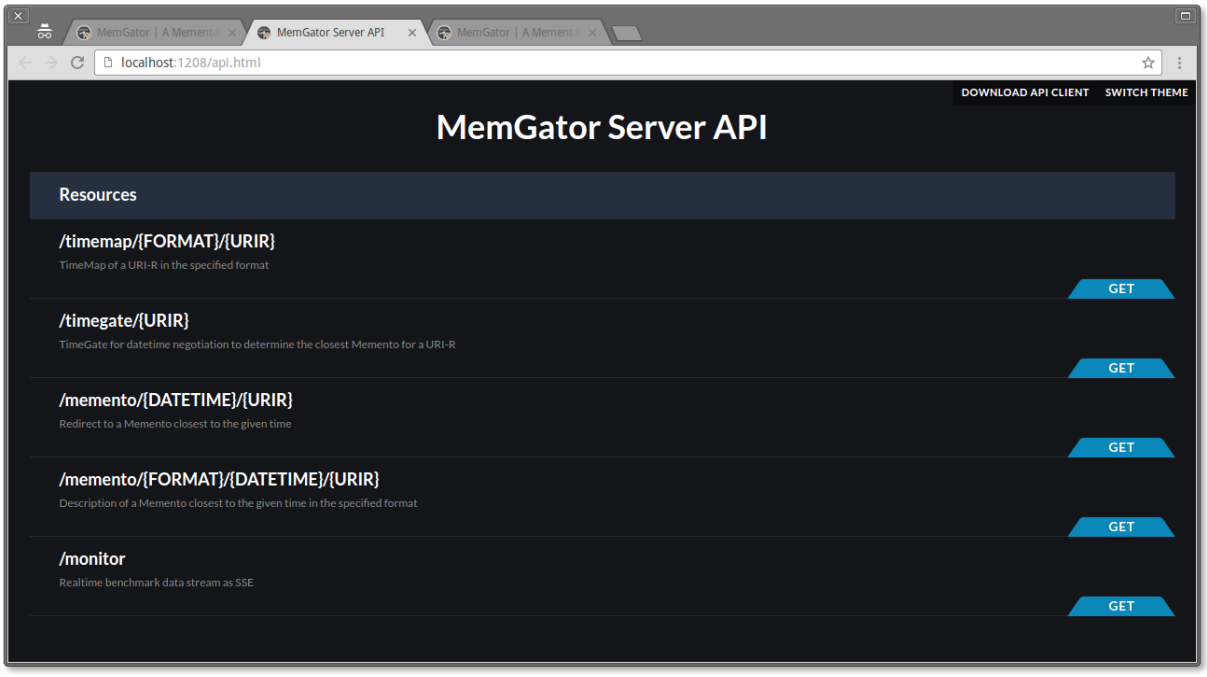
### 5.1 MemGator

MemGator[21] is an open source, easy to use, portable, concurrent, cross-platform, and self-documented Memento aggregator CLI and server tool written in Go. MemGator implements all the basic features of a Memento aggregator (e.g., TimeMap and TimeGate) and gives the ability to customize various options including which archives are aggregated. It is being used heavily by tools and services such as Mink[22], WAIL[23], OldWeb.today (NetCapsule[24]), and archiving research projects and has proved to be reliable even in conditions of extreme load. Figure 4 show various command line options in MemGator and the interactive service API when run in the server mode. MemGator being a standalone cross-platform binary enabled some interesting use cases such as packaging it in other archiving tools or providing a handy tool for web archiving researchers. MemGator was

[21] https://github.com/oduwsdl/memgator
[22] http://matkelly.com/mink/
[23] http://machawk1.github.io/wail/
[24] https://github.com/ikreymer/netcapsule




(a) CLI Options




(b) Server API

Figure 4: MemGator: A Memento Aggregator CLI and Server in Go

architected from ground with archive profiles in mind, hence it has built-in capabilities to utilize archive profiles.

### 5.2 Archive Rank Aggregator

As part of the MemGator architecture, a separate service was introduced that is responsible to consuming various archive profiles and return a rank ordered list of archives with probabilities of finding mementos of a given URI-R in the listed archives. MemGator or any other tool can then use that list to select top-K archives to query from. This modular approach enables more use cases of archive profiles. This is still a work in progress and not yet publicly released.

### 5.3 Reddit Links/Posts Archive

An archive of 200 million Reddit links/posts posted in last 11 years. The data needs some cleanup before making it publicly available, but the script used to scrape the data from Reddit API is made available[25]. Additionally, the script is also available as a Docker image[26].

### 5.4 ORS and CDXJ Formats

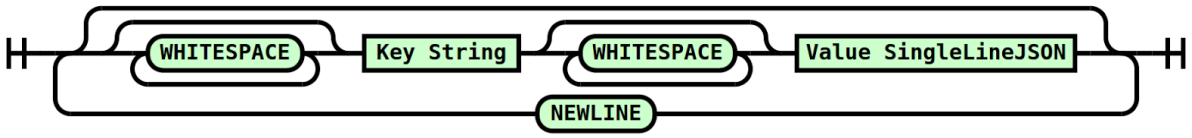
We evaluated various existing data formats for the serialization and dissemination needs of archive profiles, experimented with formats like JSON (and other similar data formats such as XML or YAML), and documented advantages and disadvantages of various formats. We ended up standardizing a new format ORS/CDXJ[27] that suits the archive profiling needs as well as proves useful in other aspects of web archiving such as archive indexes. This is a mix of CDX and line oriented JSON. This fusion of the two formats enables fast lookup, incremental updates, and arbitrary split capabilities while providing the flexibility of JSON format. Figures 5(a) and 5(b) illustrate the grammar of ORS and CDXJ respectively. Ilya Kreymer (Rhizome) contributed to the discussion about CDXJ profile serialization format.

It is worth noting that the CDXJ format has proven to be useful for various other archiving related projects. It is being used as the prefered index format in PyWB[28] and actively being pushed for the upcoming OpenWayback 3. The OWB team is defining the terms and fields for the CDXJ format that will be included in its index[29].

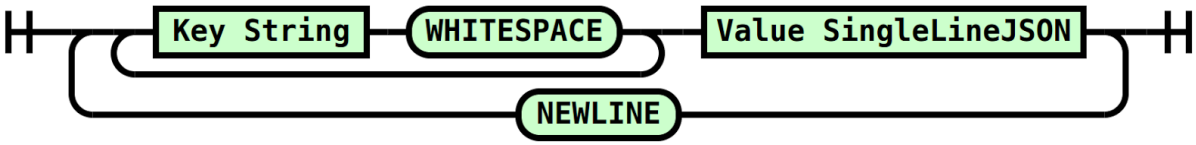
### 5.5 Binary Classifier

An approach, unanticipated at the time of the original proposal, to solve the Memento routing problem by building binary classifiers for various archives was developed, evaluated, and finally deployed in production at LANL's Time Travel service[30]. This approach utilizes the aggregator cache and does not rely on data provided by the archives, but instead builds profiles based on the history of responses for URI lookups from the archives. In summary, instead of attempting to map out an archive's contents through URIs and/or keywords tailored to the archive, it simply

[25] `https://github.com/ibnesayeed/reddit-scraper`
[26] `https://hub.docker.com/r/ibnesayeed/reddit-scraper/`
[27] `http://ws-dl.blogspot.com/2015/09/2015-09-10-cdxj-object-resource-stream.html`
[28] `https://github.com/ikreymer/pywb/wiki/CDX-Server-API`
[29] `https://iipc.github.io/warc-specifications/specifications/cdx-format/openwayback-cdxj/`
[30] `http://timetravel.mementoweb.org/`




(a) ORS Grammar




(b) CDXJ Grammar

Figure 5: Railroad diagrams of ORS and CDXJ

looks at the archive's response history based on incoming requests and uses machine learning techniques to build light-weight classifiers for routing future requests (in practice, for new archives the aggregator's cache/classifier is "warmed up" with an extensive list of URIs derived from the aggregator logs themselves). Although it was not part of the original proposal, the approach fits nicely in the usage profiling (as opposed to the content profiling). The approach is low cost and very effective for routing requests. The full description and evaluation of this approach was presented at JCDL 2016 [18].

### 5.6 Alternative TimeMap Approaches

Based on our evolving experience running both production and research archive aggregators, we saw the need for different approaches for constructing Memento TimeMaps (machine-readable lists of the URIs of mementos and their associated time of capture (Memento-Datetime)). The original approach (which is still supported) for an aggregator's TimeMap is to 1) query all archives, and 2) combine all the responses, sorted by Memento-Datetime. This approach is expensive because all archives need to be queried and their results sorted. Sometimes this full TimeMap is required and the associated expense justified, but sometimes a quick, incomplete response is sufficient. To this end, the "Do-it-yourself" (DIY) approach to TimeMaps was introduced[31]. These TimeMaps are a collection of links to TimeMaps for the requested URI at N different archives. The links are provided based on the machine learning approach described above, and as such, some listed TimeMap URIs may actually not really exist (false positives) and TimeMap URIs may exist that are not listed (false negatives). The machine learning approach leans towards avoiding the latter. Responses should be fast but the client is left with the task of collecting information about actual Mementos across archives by dereferencing the listed TimeMap URIs.

[31] `http://timetravel.mementoweb.org/guide/api/#timemap-diy`

## 6 Obstacles

Our main machine at ODU where we performed all the profiling data analysis was down for several months due to various successive disk failures. Replication of about 400 terabytes of data was not possible as no other storage was available that could accommodate it. This issue delayed some analysis such as language and hybrid profiles. However, this analysis will be published in upcoming papers as well as in Sawood's PhD dissertation. We thank Kris Carpenter (Internet Archive), Jefferson Bailey (Internet Archive), Lazarus Boone (Old Dominion University), and Joseph E. Ruettgers (Old Dominion University) for helping us with the Archive-It datasets.

## References

[1] Sawood Alam. Archive Profile Serialization. presentation at the IIPC GA 2015, slides at `http://www.cs.odu.edu/~salam/presentations/iipc/serialization.html`.

[2] Sawood Alam. Archive X-Ray - Web Archive Profiling for Efficient Memento Aggregation. presentation at ODU PhD Gathering 2015, slides at `http://www.cs.odu.edu/~salam/drafts/PhDGatheringArchiveXRay.pdf`.

[3] Sawood Alam. Archive X-Ray - Web Archive Profiling for Efficient Memento Aggregation. presentation at ODU HPC Day 2016, slides at `http://www.cs.odu.edu/~salam/presentations/HPCDayArchiveXRay.pdf`.

[4] Sawood Alam. MemGator A Memento Aggregator CLI and Server in Go. presented in Archives Unleashed 2.0: Web Archive Datathon, June 14 15, 2016, `http://ws-dl.blogspot.com/2016/06/2016-06-27-archives-unleashed-20-web.html`.

[5] Sawood Alam. MemGator A Memento Aggregator CLI and Server in Go. presented in Archives Unleashed: Web Archive Hackathon, March 3 5, 2016, `http://ws-dl.blogspot.com/2016/03/2016-03-07-archives-unleashed-web.html`.

[6] Sawood Alam. Profiling Web Archives. presentation at the IIPC GA 2015, slides at `http://www.cs.odu.edu/~salam/presentations/iipc/`.

[7] Sawood Alam. Profiling Web Archives. presentation at the WADL 2015, slides at `http://www.slideshare.net/ibnesayeed/profiling-web-archives-49856525`.

[8] Sawood Alam. Web Archive Profiling for Efficient Memento Aggregation. doctoral consortium presentation at JCDL 2016, `http://ws-dl.blogspot.com/2016/06/2016-06-30-jcdl-2016-doctoral.html`.

[9] Sawood Alam. Web Archive Profiling for Efficient Memento Aggregation. doctoral consortium presentation at TPDL 2016.

[10] Sawood Alam. Web Archive Profiling for Efficient Memento Aggregation. in progress dissertation at ODU, expected to be completed by 2017.

[11] Sawood Alam, Ilya Kreymer, and Michael L. Nelson. Object Resource Stream (ORS) and CDX-JSON (CDXJ) Formats. specification drafts at `https://github.com/oduwsdl/ORS`.

[12] Sawood Alam and Michael L. Nelson. MemGator - A Portable Concurrent Memento Aggregator. demo and poster at JCDL 2016, http://www.cs.odu.edu/~salam/drafts/memgator-jcdl16-poster.pdf.

[13] Sawood Alam and Michael L. Nelson. MemGator - A Portable Concurrent Memento Aggregator demo, poster, and lightening talk at WADL 2016.

[14] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S.H. Rosenthal. Profiling Web Archives - For Efficient Memento Query Routing. TCDL Bulletin 2015 (Proceedings of WADL 2015), http://www.ieee-tcdl.org/Bulletin/v11n2/papers/alam.pdf.

[15] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S.H. Rosenthal. Web Archive Profiling Through CDX Summarization. TPDL 2015, doi:10.1007/978-3-319-24592-8_1, http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf.

[16] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S.H. Rosenthal. Web Archive Profiling Through CDX Summarization. IJDL 2016, doi:10.1007/s00799-016-0184-4.

[17] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, and David S.H. Rosenthal. Web Archive Profiling Through Fulltext Search. TPDL 2016, http://www.cs.odu.edu/~mln/pubs/tpdl-2016/tpdl-2016-alam.pdf.

[18] Nicolas J. Bornand, Lyudmila L. Balakireva, , and Herbert Van de Sompel. Routing Memento Requests Using Binary Classifiers. JCDL 2016, http://arxiv.org/abs/1606.09136.