Evaluating Sliding and Sticky Target Policies by Measuring Temporal Drift in Acyclic Walks Through a Web Archive

Scott G. Ainsworth
Old Dominion University
Norfolk, VA, USA
sainswor@cs.odu.edu

Michael L. Nelson
Old Dominion University
Norfolk, VA, USA
mln@cs.odu.edu

ABSTRACT

When a user views an archived page using the archive's user interface (UI), the user selects a datetime to view from a list. The archived web page, if available, is then displayed. From this display, the web archive UI attempts to simulate the web browsing experience by smoothly transitioning between archived pages. During this process, the target datetime changes with each link followed; drifting away from the datetime originally selected. When browsing sparsely-archived pages, this nearly-silent drift can be many years in just a few clicks. We conducted 200,000 acyclic walks of archived pages, following up to 50 links per walk, comparing the results of two target datetime policies. The Sliding Target policy allows the target datetime to change as it does in archive UIs such as the Internet Archive's Wayback Machine. The Sticky Target policy, represented by the Memento API, keeps the target datetime the same throughout the walk. We found that the Sliding Target policy drift increases with the number of walk steps, number of domains visited, and choice (number of links available). However, the Sticky Target policy controls temporal drift, holding it to less than 30 days on average regardless of walk length or number of domains visited. The Sticky Target policy shows some increase as choice increases, but this may be caused by other factors. We conclude that based on walk length, the Sticky Target policy generally produces at least 30 days less drift than the Sliding Target policy.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

Keywords

Digital Preservation, HTTP, Resource Versioning, Temporal Applications, Web Architecture, Web Archiving

JCDL'13, July 22–26, 2013, Indianapolis, Indiana, USA.

ACM 978-1-4503-2077-1/13/07 ...$15.00.

1. INTRODUCTION

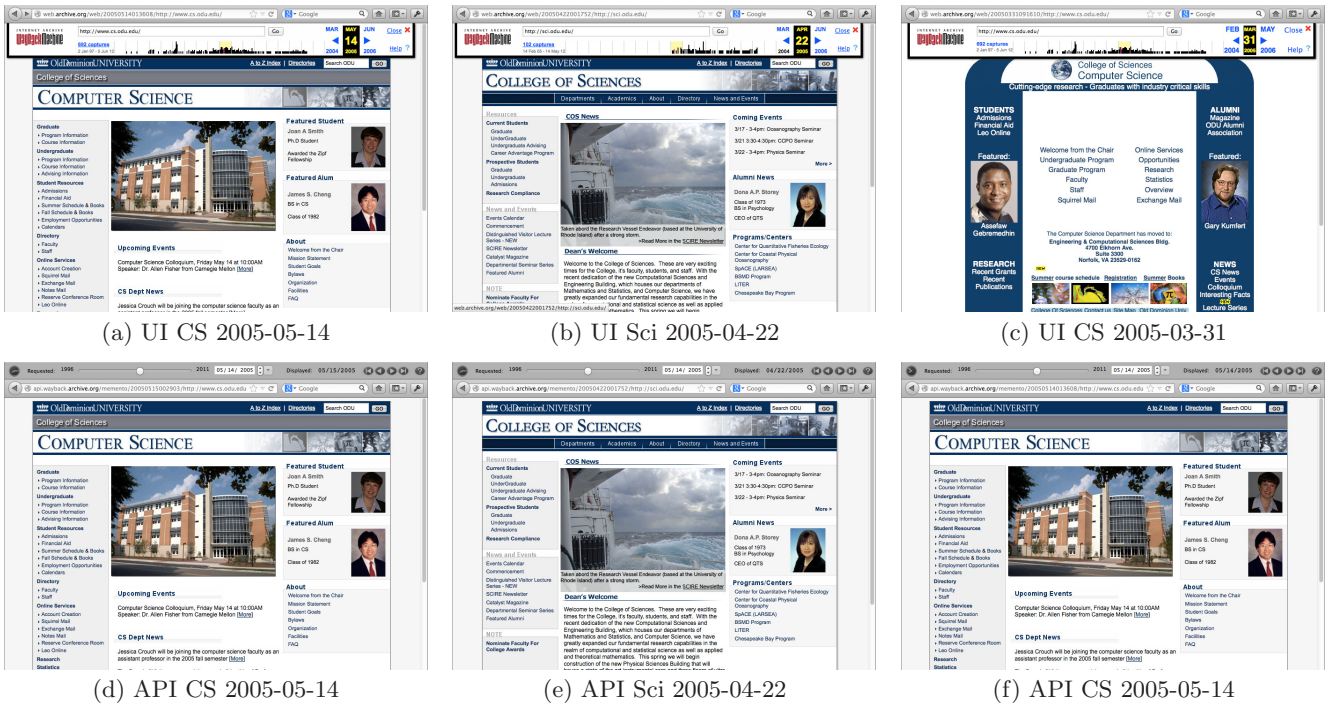
To browse archived pages from a web archive such as the Internet Archive [19], the user begins with the selection of a URI followed by selection of a Memento-Datetime (the datetime the resource was archived). Following these selections, the user is able to browse the archive's collection of mementos (archived copies) by clicking links on displayed pages; a process similar to browsing the live Web. However, with each click, the target datetime (the datetime requested by the user) is changed to the Memento-Datetime of the displayed page. Although this constant change is visible in the web browser address bar and the archive's user interface (UI), the change is easy to overlook because the change happens without explicit user interaction.

The screen shots in the top row of Figure 1 illustrates a clear case of this phenomenon. Archives of the Old Dominion University Computer Science and College of Sciences home pages are shown. The process begins by entering `http://www.cs.odu.edu` in the Internet Archive's archive browser, The Wayback Machine. The user is then presented with a list of archive datetimes, from which May 14, 2005 01:36:08 GMT is selected. The user views the Computer Science home page [Figure 1(a)]. The page URI is `http://web.archive.org/web/20050514013608/http://www.cs.odu.edu/`; note that the datetime encoded[1] in the URI matches the date selected. When the user clicks the *College of Sciences* link, the page is displayed [Figure 1(b)]. However, the encoded datetime changed to 20050422001752, a drift of 22 days. This datetime also becomes the new target datetime. When the user clicks the *Computer Science* link, the result is a different version than first displayed, as shown in Figure 1(c).

On the other hand, using a Memento-enabled browser, such as Firefox with the MementoFox add-on [20], maintains a consistent target datetime as the user follows links. The bottom row of Figure 1 shows the results. Using the API, each visit to the Computer Science home page returns the same version [Figures 1(d) and 1(f)]. The rough statistics in Table 1 show the potential improvement that can be achieved using the Memento API.

The simple example above raises many questions. How much drift do users experience when browsing archives using user interfaces such as the Wayback Machine? If the Memento API is used instead, how much drift is experienced? Which method is better and by how much? What factors contribute, positively or negatively, to the amount of drift? In particular, does the number of links available

[1]Date and time formatted YYYYMMDDHHMMSS



(a) UI CS 2005-05-14 (b) UI Sci 2005-04-22 (c) UI CS 2005-03-31

(d) API CS 2005-05-14 (e) API Sci 2005-04-22 (f) API CS 2005-05-14

Figure 1: Impact of Drift on Archive Browsing

Table 1: Temporal Drift Example

	Wayback Machine UI		Memento API	
Page	Datetime	Drift	Datetime	Drift
CS Home	2005-05-14	–	2005-05-14	–
Sci Home	2005-04-22	22 days	2005-04-22	22 days
CS Home	2005-03-31	44 days	2005-05-14	0 days
Mean		33 days		11 days

(choice), number of domains visited, or the number of links followed (walk length) contribute to drift?

2. RELATED WORK

Although the need for web archiving has been understood since nearly the dawn of the Web [8], these efforts have been for the most part independent in motivation, requirements, and scope. The Internet Archive, the first archive to attempt global scope, came into existence in 1995 [15]. Since then, many other archives have come into existence. Some of these use software developed by the Internet Archive and have similar capture behavior and user interfaces; however, other archives such as WebCite [12] have significantly different capture behaviors.

Large-scale web archiving requires resolution of issues and approaches on several axes. Although somewhat out of date, Masanès [16] is an excellent introduction. Masanès covers a broad range of web archiving topics. Of significance to this research are the technical aspects of acquisition, organization and storage, and quality and completeness. A major area not addressed by Masanès is access to archives, in particular the lack of standards or conventions for accessing archived resources. Van de Sompel et al. [26] addressed this lack with Memento.

2.1 Acquisition

Acquisition is the technical means of bringing content into an archive. Client-side archiving essentially emulates web users following links, obtaining content using the HTTP protocol. The Heritrix [18] crawler and the mirroring capability of *wget*[2] are examples of client-side archiving. A significant issue with client-side archiving is that only those parts of the Web exposed as linked resources are captured. Transactional archiving is specifically designed to overcome this limitation. Transactional archiving [7, 11, 13] inserts the capture process between the user and the data source, for example an Apache web server filter, which requires the cooperation of the server operator. Unique request-response pairs are archived, including requests for resources that are not linked. Server-side archiving makes a direct copy of the content from the server, bypassing HTTP altogether. Although conceptually simple, access to the resulting server-side archive can be difficult, requiring different URIs and navigational structures than the original. Many systems, e.g. content management systems and wikis, perform server-side archiving by design.

2.2 Organization and Storage

Once acquired content must be stored. Masanès [16] describes three organization and storage methods that are commonly used. Local archives store content in the local file system, transforming the content just enough to allow off-line browsing. Links must be modified to reference either locally-stored archived resources or the live web. Strict adherence to the original content is generally impractical and size is limited by local storage capacity and speed. Thus, local archives are most suitable for small-scale archiving. A

[2] `http://www.gnu.org/software/wget/`

common method of creating local archives is *wget* mirroring. Web-served archives, like the IA, commonly store content in WARC (Web ARChive) container files, which allows the original content and URIs to be stored unmodified. This also overcomes many limitations imposed by file systems. Content is provided to users over HTTP. Web-served archiving is highly scalable and suitable for large-scale archiving. Non-web archives generally transform web content into other forms. For example, Adobe Acrobat has the ability to download web content and produce a corresponding PDF. This type of archiving is generally best suited for resources, such as digitized books, originally created independently from the Web. Of the three types of organization and storage methods, only web-served archives are relevant to this study.

2.3 Access

An area of web archives that remained unresolved until recently was lack of methods or a standard API for time-based access to archived resources. Each archive provides a user interface (UI) to access the archive's resources. (Many archive's use the Internet Archive's Wayback Machine [24] and therefore share similar UIs.) In general, UI access to archives starts with a user-selected URI and datetime, after which the archive allows the user to simply click links to browse the collection. UI archive access is addressed in greater detail in Section 3.1.

Van de Sompel et al. addressed the lack of a standard API with Memento [25, 26], an HTTP-based framework that bridges web archives with current resources. It provides a standard API for identifying and dereferencing archived resources through datetime negotiation. In Memento, each original resource, URI-R, has zero or more archived representations, URI-M_i, that encapsulate the URI-R's state at times t_i. Using the Memento API, clients are able to request URI-M_i for a specified URI-R by datetime. Memento is now an IETF Internet Draft [25]. Memento archive access is addressed in greater detail in Section 3.2.

2.4 Quality and Completeness

In general, quality is functionally defined as fitting a particular use and objectively defined as meeting measurable characteristics. This examination of web archive content is concerned with the latter. For web archives, most quality issues stem from the difficulties inherent in obtaining content using HTTP [16]. Content is not always available when crawled, leaving gaps in the coverage. Web sites change faster than crawls can acquire their content, which leads to temporal incoherence. Ben Saad et al. [6] note that quality and completeness require different methods and measures *a priori* or *a posterior*, that is during acquisition or during post-archival access respectively.

2.4.1 Completeness (Coverage)

When crawling to acquire content, the tradeoffs required and conditions encountered lead to incomplete content or coverage. A web archive may not have the resources to acquire and store all content discovered. Associated compromises include acquiring only high priority content and crawling content less often. The content to be acquired may not be available at crawl time due to server downtime or network disruption. The combination of compromises and resource unavailability create undesired, undocumented gaps in the archive.

Although much has been written on the technical, social, legal, and political issues of web archiving; little detailed research has been conducted on the archive coverage provided by the existing archives. Day [9] surveyed a large number of archives as part of investigating the methods and issues associated with archiving. Day however does not address coverage. Thelwall touches on coverage when he addresses international bias in the Internet Archive [23], but does not directly address how much of the Web is covered. McCown and Nelson address coverage [17], but their research is limited to search engine caches. Ben Saad et al. [5, 4] address qualitative completeness through change detection to identify and archive important changes (rather than simply archiving every change). This research primarily addresses *a priori* completeness. *A posteriori* web archive coverage is addressed by Ainsworth et al. [1]. Leveraging the Memento API and pilot infrastructure, Ainsworth et al. [1] obtained results showing that 35–90% of publicly-accessible URIs have at least one publicly-accessible archived copy, 17–49% have two to five copies, 1–8% have six to ten copies, and 8–63% at least ten copies. The number of URI copies varies as a function of time, but only 14.6–31.3% of URIs are archived more than once per month. The research also shows that coverage is dependent on social popularity.

2.4.2 Temporal Coherence

When crawling to acquire content, tradeoffs are required. Crawling consumes server resources, thus crawls must be polite, e.g. paced to avoid adversely impacting the server. The web archive may not have the bandwidth needed to crawl quickly. These and other constraints increase crawl duration, which in turn increases the likelihood of temporal incoherence.

Spaniol et al. [21] note that crawls may span hours or days, increasing the risk of temporal incoherence, especially for large sites, and introduces a model for identifying coherent sections of archives, which provides a measure of quality. Spaniol et al. also present a crawling strategy which helps minimize incoherence in web site captures. In a separate paper, Spaniol et al. [22] also develop crawl and site coherence visualizations. Spaniol's work, while presenting an *a posteriori* measure, concerns the quality of entire crawls.

Denev et al. present the SHARC framework [10], which introduces a stochastic notion of *sharpness*. Sites changes are modeled as Poisson processes with page-specific change rates. Change rates can differ by MIME type and depths within the site. This model allows reasoning on the expected sharpness of an acquisition crawl. From this they propose four algorithms for site crawling. Denev's work focuses on *a priori* quality of entire crawls and does not address the quality of existing archives and crawls.

Ben Saad et al. [6] address both *a priori* and *a posteriori* quality. Like Denev et al. [10], the *a priori* solution is designed to optimize the crawling process for archival quality. The *a posteriori* solution uses information collected by the *a priori* solution to direct the user to the most coherent archived versions.

All of the above research shares a common thread: evaluation and control of completeness and temporal coherence during the crawl with the goal of improving the archiving process. In contrast, our research takes a detailed look at the quality and use of existing archives.

3. BROWSING AND DRIFT

Fundamentally, drift is the difference between the target datetime originally required and the Memento-Datetime returned by an archive. Drift can be forward or backward in time; in this study only the amount of drift is relevant. This paper examines two target datetime policies:

- **Sliding Target**: the target datetime changes as the user browses. The Memento-Datetime of the current page becomes the new target datetime.
- **Sticky Target**: the target datetime is set once at the beginning of the browsing session.

3.1 Sliding Target (The Wayback Machine)

Browsing using the Internet Archive's Wayback Machine User Interface (UI) employs the *Sliding Target* datetime policy. This policy has the potential to introduce permanent drift at every step. Here is the browsing process in detail:

1. **Select URI-R**. Navigate to `http://www.archive.org` and enter a URI-R. Clicking the *Take Me Back* button displays a calendar of the most recent year for which the URI-R is archived. The 2005 calendar for the ODU Computer Science home is shown in Figure 2.

2. **Select Memento-Datetime**. Dates covered by blue circles have mementos for the URI-R. Larger circles indicate multiple mementos for the same date. Hovering over a circle pops up a box that allows mementos to be selected. When a memento is selected, its Memento-Datetime becomes the target datetime and the corresponding memento it is displayed (as was previously shown in Figure 1). Drift is introduced when the selected memento redirects to another memento that has a different Memento-Datetime than originally selected.

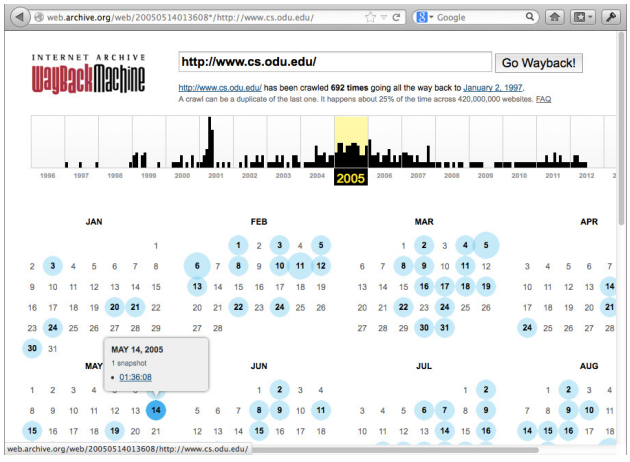


Figure 2: Wayback Machine Calendar

3. **Browse additional URI-Rs**. To simulate browsing the Web within the context of the archive, links are rewritten to reference the archive instead of the original URI and to embedded the Memento-Datetime of the displayed memento. Each link followed uses the embedded datetime as the new target datetime (the selection from step 2 is forgotten), which introduces drift. Additionally, it is unlikely that the selected URI-R was archived at the new target datetime; therefore, one or more additional redirects, each introducing drift, will be required before the best memento is displayed.

Browsing using the *Sliding Target* policy introduces two kinds of drift: *Memento drift* by redirection to the actual memento and *Target drift* by changing the target datetime.

3.2 Sticky Target (Memento API)

Browsing the Internet Archive using the Memento API uses the Sticky Target policy. The Sticky Target policy also introduces drift; however, the drift is constrained because the target datetime is maintained. Here is the browsing process using Firefox and the MementoFox add-on:

1. **Select URI-R**. Open Firefox and enable MementoFox. Move the requested datetime slider to the desired target datetime. All URI-Rs entered in the address bar or followed through clicking a link, are now dereferenced using the Memento API and redirected to the *best* URI-M, which is the URI-M with Memento-Datetime closest to the target datetime. Figure 3 shows the ODU Computer Science home for 2005-05-15T00:28:03Z as dereferenced by MementoFox. Drift is introduced when the target datetime redirects to a memento with a Memento-Datetime that is not the target datetime.

Figure 3: Memento API and MementoFox

2. **Browse additional URI-Rs**. Each subsequent link clicked uses the target datetime selected by the slider. Drift continues to be introduced by redirects as in step 1; however, using MementoFox and the Memento API allows the target datetime to remain the same for every request.

Thus, browsing using the Memento API introduces only memento drift and does not introduce target drift.

4. EXPERIMENT

4.1 Samples

Building on our previous coverage work, we use the same four URI sample sets (DMOZ, Delicious, Bitly, and Search

Engines) as in [1]. Each sample contains 1,000 randomly-selected URIs for 4,000 URI total. URI random selection details can be found in [2].

Table 2: Archival Rates

Sample	2013	2011
DMOZ	95.2%	79%
Delicious	91.9%	68%
Bitly	23.5%	16%
Search Engine	26.4%	19%
Aggregate	59.4%	46%

Table 2 shows the percentage of URI-Rs in the sample that were found to be archived during the experiment. There are several notable differences from our 2011 results [27]. The DMOZ and Delicious samples are archived at a considerably higher rate; the Bitly and Search Engine samples rate are only slightly higher. We attribute this to the increased archive availability provided by the Internet Archive over the past two years [14].

4.2 Procedure

The examination of temporal drift was accomplished in January 2013. To ensure adequate number of successful walks, 200,000 walks were attempted. Each walk was accomplished in three phases:

- Obtain the initial memento,
- Follow links, and
- Calculate drift and statistics.

Each walk iterates through the process of selecting URI-Rs and downloading mementos until either 50 steps are successful or an error is encountered. The details of each walk step are captured, including steps that encounter errors. The last step will contain the stop reason, unless the walk completed 50 successful steps (in which case there is no stop reason). The vast majority of walks encounter an error before reaching step 50. The length of a walk is the number of successful steps. For example, a walk that stops on walk step 6 (i.e. step 6 encounters an error), is length 5 because the first 5 steps were successful. Table 3 defines the symbols used in the procedure description.

To ensure repeatability, a set of 200,000 random numbers (one per walk) were generated. These random numbers were used as both the walk identifier and as the seed to initialize the walk's random number generator. The experiment was run under Apple OS X 10.8.2 using Python 2.7.2 (the version included with OS X). The built-in Python random number generator, `random.Random`, was used. Resources were downloaded using *curl*, which is much more robust than the Python libraries.

Phase I. First Walk Step

Phase I selects a walk's first URI-R, downloads the corresponding timemap, and downloads the first API and UI mementos. Phase I accomplishes the first step of a walk.

1. Randomly select R_1 from the 4,000 sample URIs.
2. Retrieve the timemap for R_1 from the Internet Archive using the Memento API.

Table 3: Definitions

Term	Definition
W	An acyclic walk.
w	An acyclic walk index.
i	A walk step index.
t	The target datetime. t_i is the target for walk step i.
R	A URI-R. R_i is the R selected for walk step i.
M	A URI-M.
M^a	A Memento API URI-M. M_i^a is the M^a for walk step i.
M^w	A Wayback Machine UI URI-M. M_i^w is the M^w for walk step i.
L	The set of link URI-Rs in a memento (dereferenced URI-M).
$\mathcal{T}(M)$	The Memento-Datetime of M.
$\mathcal{L}(M)$	The set of link Rs in the memento identified by M.
$\Delta(M)$	The drift of M relative to the corresponding t. $\Delta(M_i) = \lvert t_i - \mathcal{T}(M_i)\rvert$
δ^a	Memento API memento drift. $\delta_i^a = \Delta(M_i^a)$.
δ^w	Wayback Machine UI memento drift. $\delta_i^w = \Delta(M_i^w)$.

3. Randomly select a URI-M, M_1, from the timemap. M_1 yields this step's target datetime, $t_1 = \mathcal{T}(M_1)$.
4. Dereference M_1^a using t_1 from the IA Memento API. HTTP redirects may occur during dereferencing, yielding $M_1^{a\prime}$ as the final dereferenced URI-M. Note that $M_1^{a\prime} = M_1^a$ may not hold. It follows that $\mathcal{T}(M_1^{a\prime}) = t_1$ also may not hold. This is the *Sticky Target* policy.
5. Calculate the corresponding M_1^w and dereference it. As in step 4, HTTP redirects may occur during dereferencing, yielding $M_1^{w\prime}$ as the final dereferenced URI-M. In addition, the Wayback Machine returns *soft* redirects. These responses have HTTP status 200 but contain embedded JavaScript redirects; these are detected and followed. Note that $M_1^{w\prime} = M_1^w$ may not hold. It follows that $\mathcal{T}(M_1^{w\prime}) = t_1$ also may not hold. This is the *Sliding Target* policy.

Phase II. Additional Walk Steps

Phase II accomplishes a walk's second and subsequent steps. It selects a link common to the API and UI mementos from the previous walk step and downloads the corresponding timemap and mementos. If there are no common links, the walk stops. In the following, i is the current walk step.

6. Extract the sets of link URI-Rs, $L^a = \mathcal{L}(M_{i-1}^a)$ and $L^w = \mathcal{L}(M_{i-1}^w)$, from the previous walk step's mementos. Denote the set of URI-Rs used in previous walk steps is L^p. Then, the set of common, usable URI-Rs for this walk step is $L_i^u = (L^a \cap L^w) - L^p$. Randomly select R_i from L_i^u.
7. Download the timemap for R_i from the IA Memento API.
8. Select the *best* URI-M, M_i^a, from the timemap. M_i^a is the URI-M that minimizes $\lvert t_1 - \mathcal{T}(M)\rvert$, e.g. the URI-M nearest the initial target datetime.

9. Dereference M_i^a using t_1 from the IA's Memento API. As in step 4, HTTP redirects may occur during dereferencing, yielding $M_i^{a\prime}$. $M_i^{a\prime} = M_i^a$ and $\mathcal{T}(M_i^{a\prime}) = t_1$ may not hold. This is the *Sticky Target* policy.

10. Calculate M_i^w using $t_i = \mathcal{T}(M_{i-1}^{u\prime})$, as the target datetime. Dereference M_i^w. As in step 5, HTTP redirects and *soft* redirects may occur, yielding $M_1^{w\prime}$ as final. Again, $M_i^{w\prime} = M_i^w$ and $\mathcal{T}(M_i^{w\prime}) = t_i$ may not hold.

11. Repeat Phase II until L_i^u is empty, an error occurs, or 50 walk steps have been completed successfully.

Phase III. Drift Calculations

Phase III calculates drift and statistics, ignoring duplicate walks. Duplicate walks occurs for a number of reasons. A common reason is failure on the first walk step because R_1 has never been archived. A limited number of links or mementos is another reason.

12. Calculate API drift, $\delta_i^a = \Delta(M_i^{a\prime})$, for each successful walk step. Calculate the API drift mean, median, and standard deviation for the entire walk.

13. Calculate Wayback Machine drift, $\delta_i^w = \Delta(M_i^{w\prime})$, for each successful walk step. Calculate the Wayback Machine drift mean, median, and standard deviation for the entire walk.

4.3 Results

The 200,000 acyclic walks attempted resulted in 53,926 unique walks, of which 48,685 had at least 1 successful step. Overall there were 240,439 successful steps, with an average of 3.85 successful steps per walk. Table 4 summarizes walks and steps by sample.

Table 4: Walks and Steps

	DMOZ	S.Eng.	Delicious	Bitly	Total
Steps	64,661	26,047	124,020	25,711	240,439
Succ. Steps	48,445	20,177	98,560	20,189	187,371
w/$\delta^u > 1$yr	1,761	1,212	3,028	700	6,701
w/$\delta^u > 5$yr	75	13	16	7	111
Unique Walks	16,221	5,873	25,482	5,524	53,100
Succ. Walks	15,009	4,604	24,451	4,621	48,685
Pct. Succ.	92.5%	78.4%	96.0%	83.7%	91.7%
Mean Succ. Steps/Walk	3.2	4.4	4.0	4.4	3.8

4.3.1 Walk Length and Stop Causes

Figure 4 shows the distribution of successful walks by length. (Note that *Occurrences* is a log scale.) Table 5 shows the details behind Figure 4, broken out by sample. Walks greater than 25 in length are summarized in groups of 5. The number of steps decreases exponentially as walk length increases. Less than 1% of walks progress past step 21. For DMOZ, less than 1% progress step past 19; Search Engine, step 23; Delicious, step 23; and Bitly, step 24.

Table 6 summarizes the reasons walks stop before reaching step 50, split by timemap and memento. Because the selection processes for the first and subsequent mementos differ, separate statistics are shown. The stop causes are dominated by 403s, 404s, 503s, and *No Common Links*. The 403s

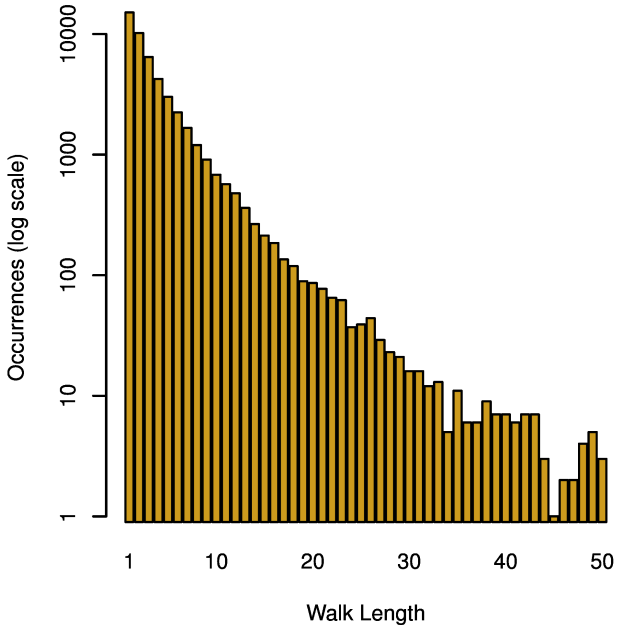


Figure 4: Occurrences by Walk Length

are generally an archived 403; the original URI-R returned a 403 when accessed for archiving. The timemap 404s indicate that the URI-R is not archived. Memento 404s can have two meanings: either the original URI-R returned a 404 when it was accessed for archiving or the memento has been redacted, i.e. removed from public access. The 503s most likely indicate a transient condition such as an unavailable archive server, thus there is a chance that on repetition the resource will be found. Resources that returned 503s were retried a single time one week after the first 503 was received. Less than 1% succeed on the second try. *Download failed* indicates that *curl* was unable to download the resource; like the 503s, these were retried once. *Not HTML* indicates that the downloaded resource was not HTML and therefore not checked for links. *No common links* indicates that although both Memento API and Wayback Machine UI mementos were found, content divergence due to drift caused the two mementos to have no common links.

4.3.2 Drift

Figure 5 illustrates the distribution of Wayback Machine mementos by drift (δ^w). The horizontal axis is the walk step number. The vertical axis is the drift in years from the walk step's target datetime. Color indicates memento density on an exponential axis. As expected, density is highest for early steps and tapers off with as walk length increases. Density is also highest at low drift values and many mementos appear to have very high drift. However, only 11,093 (4.6%) exceed 1 year and only 172 (0.07%) exceed 5 years (Table 4). It is also interesting that the first step shows drift (as high as 6.5 years). The first target datetime is selected from a timemap, which sets the expectation that a memento for the datetime exists. However, redirects (4.2 steps 4, 5, 9, and 11), from the URI-M in the timemap to the final URI-M cause drift—even on the first walk step.

Figure 6 illustrates the distribution of Memento API mementos by drift (δ^a), which at first glance appears very sim-

Table 5: Occurrences by Length

Walk Length	DMOZ	Search Engine	Delicious	Bitly	Total
1	5,355	1,239	7,193	1,289	15,076
2	3,571	924	4,857	817	10,169
3	1,891	598	3,311	623	6,423
4	1,212	381	2,228	415	4,236
5	791	315	1,588	314	3,008
6	583	232	1,168	259	2,242
7	417	178	877	186	1,658
8	258	153	651	136	1,198
9	187	111	498	108	904
10	144	79	377	79	679
11	114	71	306	74	565
12	99	51	279	48	477
13	72	44	200	46	362
14	54	32	144	35	265
15	39	30	119	26	214
16	33	26	108	20	187
17	20	23	76	18	137
18	24	14	68	12	118
19	19	12	46	12	89
20	14	10	47	15	86
21	20	11	36	9	76
22	7	13	28	16	64
23	11	11	33	7	62
24	7	4	22	4	37
25	8	3	25	3	39
26–30	27	18	68	20	133
31–35	7	7	30	14	58
36–40	6	3	23	3	35
41–45	6	2	14	2	24
46–50	6	3	6	1	16
Totals	15,002	4,598	24,426	4,611	48,637

Table 6: Stop Causes

Stop Cause	First Step Count	First Step Percent	Other Steps Count	Other Steps Percent
Timemaps				
HTTP 403	74	1.7%	4,803	9.1%
HTTP 404	1,327	30.1%	15,850	29.9%
HTTP 503	0	0.0%	43	0.1%
Other	2	0.0%	180	0.3%
Mementos				
HTTP 403	52	1.2%	476	0.9%
HTTP 404	215	4.9%	3,633	6.8%
HTTP 503	1,957	44.4%	10,535	19.9%
Download failed	154	3.5%	589	1.1%
Not HTML	514	11.7%	2,856	5.4%
No Common Links		0.0%	12,957	24.4%
Other	117	2.7%	1,128	2.1%
Totals	**4,412**		**53,050**	

ilar to Figure 5. Closer examination reveals that many mementos have lower drift when using the Memento API. Figure 7 shows the mean drift by step (solid circles) and standard deviation (hollow circles) for both the Memento API (green) and Wayback Machine URI (blue). The Memento API, which uses the Sticky Target policy, results in 40–50 days less mean drift than the Sliding Target policy. This

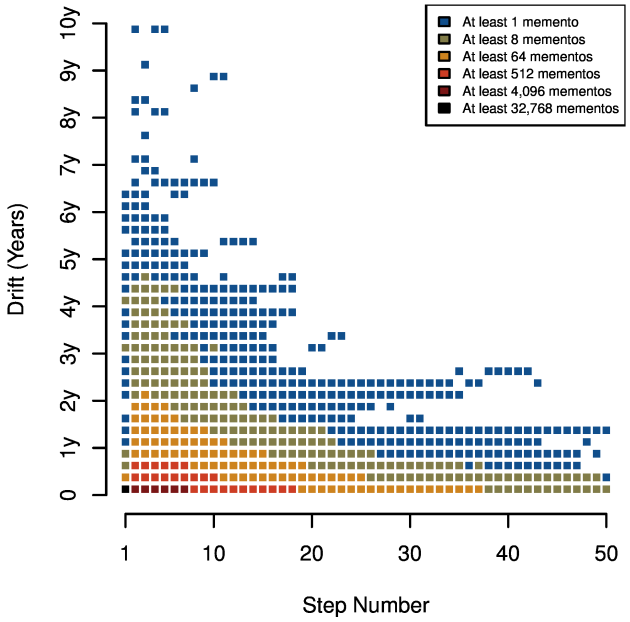


Figure 5: UI Drift by Step

delta appears to decrease above step 40, but there are only 40 walks (0.0082%) that achieve this many steps (see Table 5), so the decrease is not significant.

Even below step 40, mean as a useful indicator of central tendency is in doubt. As Figure 7 shows, the standard deviation significantly exceeds the mean, particularly at low step numbers. This indicates that median may be a better indicator of the central tendency. Note the horizontal line of squares at 1.25 years in Figures 5 and 6. Investigation revealed that well-archived, self-contained sites contribute more long walks than groups of loosely-linked sites. For example, left column of every `http://www.101celebrities.com` page includes nearly 1,000 links to pages within the site; it is a primary contributor to the horizontal line. The number of domains in a walk and their impact on the walk are discussed in 4.3.4. The median is shown in Figure 8. The median shows lower average drift than the mean because it is less impacted by the outliers. For this data, we believe median is the better measure of central tendency and will use median from here forward.

4.3.3 Choice

Every walk step has a limited number of links to choose from. Given a starting URI-R and Memento-Datetime, it is possible to represent all possible walks as a tree (4.2 step 6 disallows revisits). Choice is the number of children common to both the Memento API and Wayback Machine mementos. Figure 9 shows the median drift by choice for the Memento API and Wayback Machine UI. Clearly as choice increases, drift also increases.

4.3.4 Number of Domains

Through casual observations, we began to suspect that the number of domains accessed in a walk also impacted drift. Figure 10 graphs the relationship between the number of domains and drift. The number of domains has a significant correlation with drift, but only for the Sliding Policy.

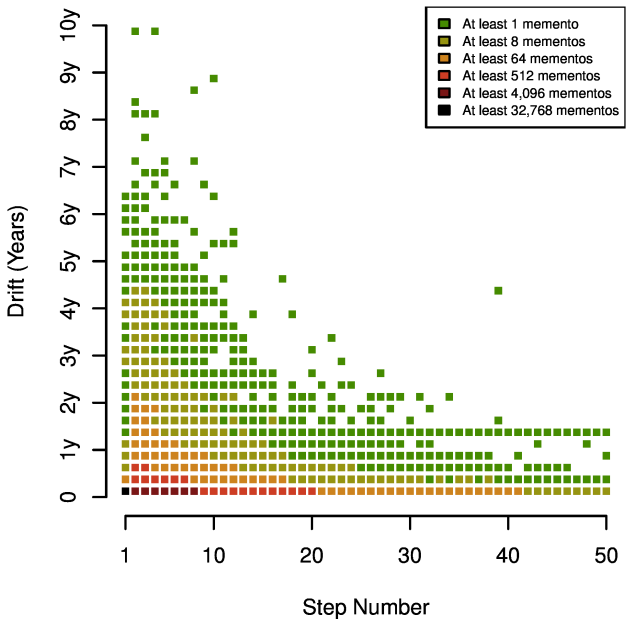


Figure 6: API Drift by Step

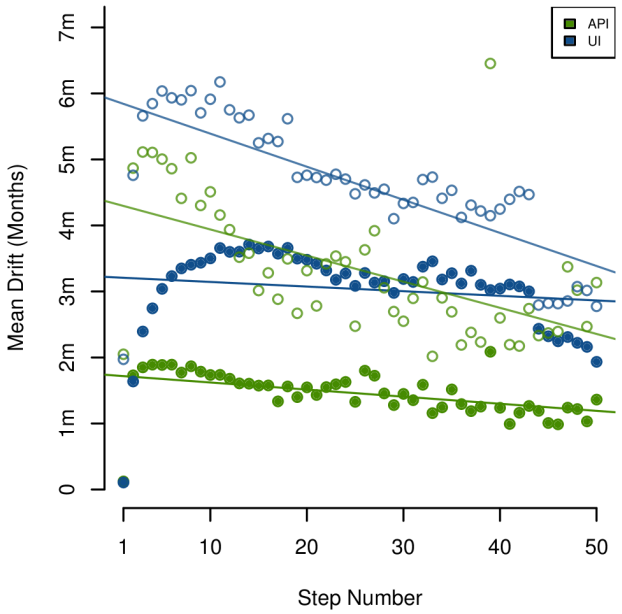


Figure 7: Standard Deviation Drift by Step

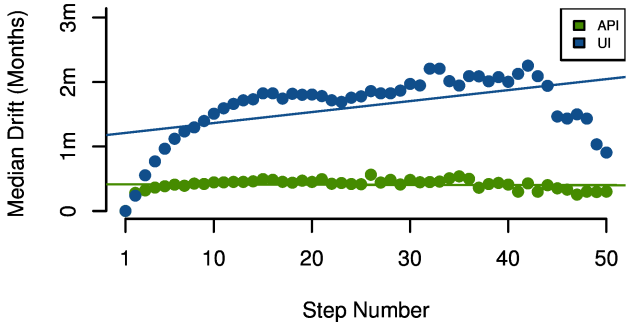


Figure 8: Median Drift by Step

4.3.5 *Sample Differences*

In our 2011 research [1], we found that archival rates varied from 16%–79% (see Table 2) depending on the sample from which the URI-R originated. This led to exploration of possible differences between acyclic walks based on sample. We found there is not a significant drift difference based on sample source.

4.3.6 *Relaxed Shared URI Requirement*

An average walk length of 3.2 steps seems short. Anecdotally, the authors' experience has been much longer walks when browsing the Internet Archive. Much of this difference is likely due to the random rather than human browsing approach [3], but questions arose about requiring common URI-Rs at every walk step (4.2 step 6). The experiment was run again using the same sample URI-Rs and random numbers while relaxing the requirement. When a common URI-R was not available, two different URI-Rs were selected. The results are summarized in Table 7. Compared with Table 4, there is little change. The number of steps and successful steps increased about 5% each. The number of unique and successful walks only increased by about 2.5% and the average number of successful steps per walk increased by only 2.3%. Figure 11 shows the median drift by step after relaxing the shared URI requirement; it is very similar to Figure 8. API drift is essentially the same and UI drift slightly reduced. Even though relaxing the shared URI requirement reduces comparability between the two policies, the results also show that the Sticky Target policy controls temporal drift and that drift grows under the the Sliding Target policy.

5. FUTURE WORK

We see several avenues of future work for this line of research. The experiments conducted so far have focused on randomly-generated walks through a single archive. Al-

Noamany et al. [3] have looked at real-world walk patterns through analysis of the Internet Archive's web server logs. Using these patterns to guide walks will provide more realistic link streams and result in temporal drift data more in line with actual user experience. There are also domains that users tend to avoid, such as link farms, SEO, and spam sites. Detecting and avoiding them, as a user would, will also move the data toward real-world user experience. We also suspect that long walk drift is heavily influenced by clusters of closely-related domains and domains that primarily self-reference. Applying an appropriate measure of clustering or similarity may shed some light on this topic.

Preliminary research has shown that the amount of drift can vary based on the starting date. Repeating this study with a focus on the earliest and latest archived versions available will bear out (or disprove) our preliminary evidence. Closely related to choosing the earliest or latest versions is starting with a variety of fixed datetimes. In this case, we hypothesize increased first step drift for early dates followed by drift settling out after a few steps.

Recently, additional archives (e.g. the UK Web Archive) have implemented native Memento support and the Wayback Machine UI. It will be interesting to see if other archives

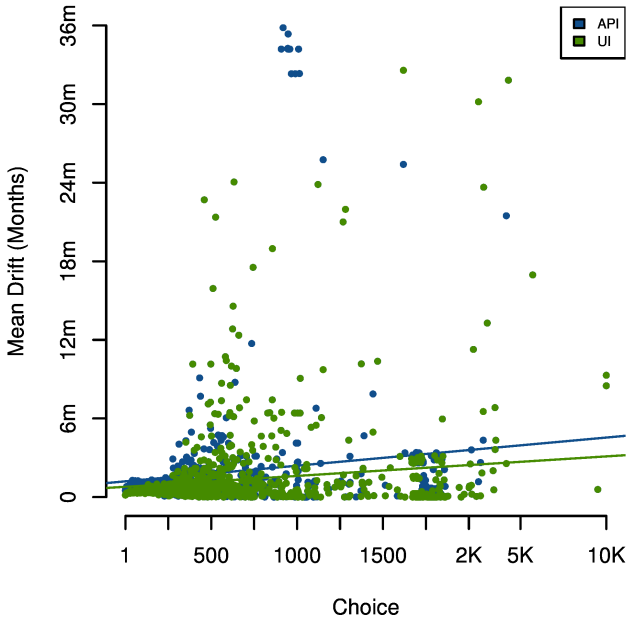


Figure 9: Median Drift by Choice

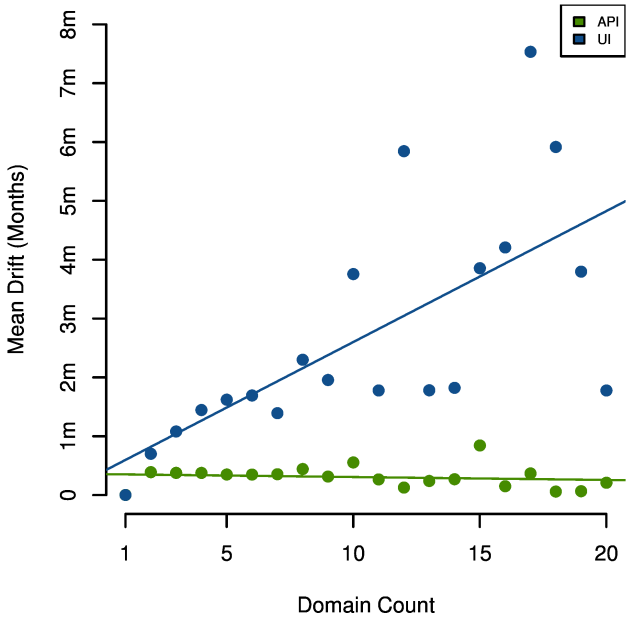


Figure 10: Median Drift by Number of Domains

have temporal drift similar to the Internet Archive's. Finally, the Memento architecture provides for aggregators, which are servers that combine the timemaps from multiple archives into a single, unified timemap. The aggregators will make it possible to study drift across multiple archives.

6. CONCLUSION

We studied the temporal drift that occurs when browsing web archives under two different policies: Sliding Target and Sticky Target. Acyclic walks through the Internet archived were conducted using the Memento API, which

Table 7: Change in Walks and Steps w/Relaxed Shared URI Requirement

	Strict	Relaxed	Change
Steps	240,439	251,439	+4.6%
Succ. Steps	187,371	196,999	+5.1%
w/$\delta^u > 1$yr	6,701	6,344	-5.3%
w/$\delta^u > 5$yr	111	118	+6.3%
Unique Walks	53,100	54,474	+2.6%
Succ. Walks	48,685	50,043	+2.8%
Pct Succ.	91.7%	91.9%	+0.2%
Successful Steps/Walk	3.2	3.9	+2.3%

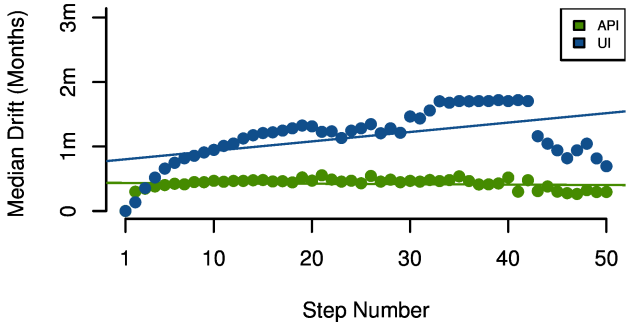


Figure 11: Median Drift by Step (Relaxed URI Requirement)

uses the Sticky Target policy, and the Wayback Machine UI, which employs the Sliding policy. Measurements of drift were taken on three axis: number of steps (Table 8), choice (Table 9), and number of domains (Table 10). All three showed a showed a positive correlation with increased temporal drift for the Sliding Target policy. For the Sticky Target policy, drift by step and drift by domain count showed no correlation. Drift by choice showed low correlation for both policies; however, median drift for the Sticky Target was still lower overall. The Sticky Target policy clearly achieves lower temporal drift. Based on walk length, the Sticky Target policy generally produces at least 30 days less drift than the Sliding Target policy.

7. ACKNOWLEDGMENTS

This work supported in part by the NSF (IIS 1009392) and the Library of Congress. We are grateful to the Internet Archive for their continued support of Memento access to their archive. Memento is a joint project between the Los Alamos National Laboratory Research Library and Old Dominion University.

8. REFERENCES

[1] S. G. Ainsworth, A. Alsum, H. SalahEldeen, M. C. Weigle, and M. L. Nelson. How much of the Web is archived? In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital libraries*, JCDL '11, pages 133–136, June 2011.

[2] S. G. Ainsworth, A. Alsum, H. SalahEldeen, M. C. Weigle, and M. L. Nelson. How much of the Web is archived? Technical Report arXiv:1212.6177, Old Dominion University, December 2012.

[3] Y. AlNoamany, M. C. Weigle, and M. L. Nelson. Access patterns for robots and humans in web archives. In *Proceedings of the 13th Annual International ACM/IEEE Joint Conference on Digital libraries*, JCDL'13, July 2013.

[4] M. Ben Saad and S. Gançarski. Archiving the Web using page changes patterns: a case study. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 113–122, 2011.

[5] M. Ben Saad and S. Gançarski. Improving the quality of web archives through the importance of changes. In *Proceedings of the 22nd international conference on Database and expert systems applications - Volume Part I*, DEXA'11, pages 394–409, 2011.

[6] M. Ben Saad, Z. Pehlivan, and S. Gançarski. Coherence-oriented crawling and navigation using patterns for web archives. In *Proceedings of the 15th international conference on Theory and practice of digital libraries: research and advanced technology for digital libraries*, TPDL'11, pages 421–433, 2011.

[7] J. F. Brunelle and M. L. Nelson. Evaluating the sitestory transactional web archive with the apachebench tool. Technical Report arXiv:1209.1811, Old Dominion University, September 2012.

[8] C. Casey. The Cyberarchive: a look at the storage and preservation of web sites. *College & Research Libraries*, 59, 1998.

[9] M. Day. Preserving the fabric of our lives: A survey of web preservation initiatives. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)*, pages 461–472, 2003.

[10] D. Denev, A. Mazeika, M. Spaniol, and G. Weikum. SHARC: Framework for quality-conscious web archiving. volume 2, pages 586–597, August 2009.

[11] C. E. Dyreson, H.-l. Lin, and Y. Wang. Managing versions of web documents in a transaction-time web server. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, 2004.

[12] G. Eysenbach and M. Trudel. Going, going, still there: Using the WebCite service to permanently archive cited web pages. *Journal of Medical Internet Research*, 7(5), 2005.

[13] K. Fitch. Web site archiving: an approach to recording every materially different response produced by a website. In *9th Australasian World Wide Web Conference, Sanctuary Cove, Queensland, Australia,*, pages 5–9, 2003.

[14] B. Kahle. Wayback machine: Now with 240,000,000,000 URLs. `http://blog.archive.org/2013/01/09/updated-wayback/`, January 2013.

[15] M. Kimpton and J. Ubois. Year-by-year: from an archive of the Internet to an archive on the Internet. In J. Masanès, editor, *Web Archiving*, chapter 9, pages 201–212. 2006.

[16] J. Masanès. Web archiving: issues and methods. In J. Masanès, editor, *Web Archving*, chapter 1, pages 1–53. 2006.

[17] F. McCown and M. L. Nelson. Characterization of search engine caches. In *Proceedings of IS&T Archiving 2007*, pages 48–52, May 2007.

[18] G. Mohr, M. Stack, I. Rnitovic, D. Avery, and M. Kimpton. Introduction to Heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop, Bath, UK*, September 2004.

[19] K. C. Negulescu. Web archiving @ the Internet Archive. `http://www.digitalpreservation.gov/news/events/ndiipp_meetings/ndiipp10/docs/July21/session09/NDIIPP072110FinalIA.ppt`, 2010.

[20] R. Sanderson, H. Shankar, S. Ainsworth, F. McCown, and S. Adams. Implementing time travel for the Web. *Code{4}Lib Journal*, (13), 2011.

[21] M. Spaniol, D. Denev, A. Mazeika, G. Weikum, and P. Senellart. Data quality in web archiving. In *Proceedings of the 3rd Workshop on Information Credibility on the Web*, WICOW '09, pages 19–26, 2009.

[22] M. Spaniol, A. Mazeika, D. Denev, and G. Weikum. "Catch me if you can": Visual analysis of coherence defects in web archiving. In *The 9 th International Web Archiving Workshop (IWAW 2009) Corfu, Greece, September/October, 2009 Workshop Proceedings*, pages 27–37, 2009.

[23] M. Thelwall and L. Vaughan. A fair history of the Web? examining country balance in the Internet Archive. *Library & Information Science Research*, 26(2), 2004.

[24] B. Tofel. 'Wayback' for accessing web archives. In *Proceedings of the 7th International Web Archiving Workshop (IWAW07)*, 2007.

[25] H. Van de Sompel, M. Nelson, and R. Sanderson. HTTP framework for time-based access to resource states — Memento, November 2010. `http://datatracker.ietf.org/doc/draft-vandesompel-memento/`.

[26] H. Van de Sompel, M. L. Nelson, R. Sanderson, L. L. Balakireva, S. Ainsworth, and H. Shankar. Memento: Time travel for the Web. Technical Report arXiv:0911.1112, 2009.

[27] M. C. Weigle. How much of the web is archived? `http://ws-dl.blogspot.com/2011/06/2011-06-23-how-much-of-web-is-archived.html`, June 2011.