Int J Digit Libr (2015) 16:283–301
DOI 10.1007/s00799-015-0150-6

Not all mementos are created equal: measuring the impact of missing resources

Justin F. Brunelle[1] · **Mat Kelly**[1] · **Hany SalahEldeen**[1] · **Michele C. Weigle**[1] · **Michael L. Nelson**[1]

Received: 3 December 2014 / Revised: 22 April 2015 / Accepted: 22 April 2015 / Published online: 6 May 2015

Abstract Web archives do not always capture every resource on every page that they attempt to archive. This results in archived pages missing a portion of their embedded resources. These embedded resources have varying historic, utility, and importance values. The proportion of missing embedded resources does not provide an accurate measure of their impact on the Web page; some embedded resources are more important to the utility of a page than others. We propose a method to measure the relative value of embedded resources and assign a damage rating to archived pages as a way to evaluate archival success. In this paper, we show that Web users' perceptions of damage are not accurately estimated by the proportion of missing embedded resources. In fact, the proportion of missing embedded resources is a less accurate estimate of resource damage than a random selection. We propose a damage rating algorithm that provides closer alignment to Web user perception, providing an overall improved agreement with users on memento damage by 17 % and an improvement by 51 % if the mementos have a damage rating delta >0.30. We use our algorithm to measure damage in the Internet Archive, showing that it is getting better at mitigating damage over time (going from a damage rating of 0.16 in 1998 to 0.13 in 2013). However, we show that a greater number of important embedded resources (2.05 per memento on average) are missing over time. Alternatively, the damage in WebCite is increasing over time (going from 0.375 in 2007 to 0.475 in 2014), while the missing embedded resources remain constant (13 % of the resources are missing on average). Finally, we investigate the impact of JavaScript on the damage of the archives, showing that a crawler that can archive JavaScript-dependent representations will reduce memento damage by 13.5 %.

Keywords Web architecture · Web archiving · Digital preservation · Memento damage

✉ Justin F. Brunelle
jbrunelle@cs.odu.edu

Mat Kelly
mkelly@cs.odu.edu

Hany SalahEldeen
hany@cs.odu.edu

Michele C. Weigle
mweigle@cs.odu.edu

Michael L. Nelson
mln@cs.odu.edu

[1] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

1 Introduction

Web archives are valuable cultural repositories that capture and store Web content. People (and robots) use archives like the Internet Archive [27,43] to retrieve archived material [20,25] for a variety of purposes and in a variety of ways [2]. However, the resources being requested by Web users may not be complete; embedded resources are sometimes missing from an archived Web page [6]. Missing embedded resources return a non-200 HTTP status (e.g., 404, 503) when their URI is dereferenced.

Archivists work to ensure archives are as complete—and as high quality—as possible. Through identifying sources of missing content or archival difficulties, archivists can address archival challenges by taking steps to adjust processes or to fill in gaps in archive collections.

Reyes et al. identified current efforts within several archives to assess their archival collections [4]. Of the archivists sampled, 61 % confirmed that their goal is to assess the quality of every Web page captured, 43 % assess quality

and success using a simple Boolean or numerical notion of completeness based upon the number of missing embedded resources in the Web pages. As we will demonstrate in this paper, human perception of quality is not accurately represented with a measure of the proportion of missing embedded resources. For example, large images are often more important to an archived page's utility than small images. Similarly, style sheets that format visible content are more important to the representation of the page than style sheets without significant formatting responsibilities. We provide a mechanism to assess the impact of missing embedded resources on the archives that improves upon simply measuring the percent of missing embedded resources.

Of the archivists surveyed by Reyes et al. that conduct quality assurance, 100 % use a manual process. The Internet Archive alone boasts 455 billion Web pages in its archive,[1] which is far larger than can be evaluated through human methods. While Banos et al. constructed the CLEAR method to assign a predictive archivability score [6], a similar score for the actual performance of an archival tool does not exist outside of the simple metric of the percent of embedded resources archived. An algorithm to automatically assess human perception of archived page quality would significantly decrease the necessary human involvement in the quality assurance process, potentially increasing the accuracy while reducing the cost of quality assurance efforts.

Throughout this paper, we use memento framework terminology. Memento [44] is a framework that standardizes Web archive access and terminology. Original (or live Web) resources are identified by URI-R, and archived versions of URI-Rs are called *mementos* and are identified by URI-M. Memento TimeMaps are machine-readable lists of mementos (at the level of single archives or aggregation of archives) sorted by archival date.

This research has three goals. First, we want to understand how missing embedded resources impacts Web users' perceived quality of a memento. Using an algorithm to measure embedded resource importance, we determine whether an important embedded resource of the memento is missing (e.g., a main image or video essential to the user's understanding of the page) or whether the missing embedded resource contributes little to the memento's utility for the user (such as a spacer image or small logo). We propose a method of weighting embedded resources in a memento according to importance (D_m). We show that D_m is an improved damage rating over an unweighted proportion of missing embedded resources to all requested resources (M_m). We use Amazon's Mechanical Turk to compare our algorithm to Web users' notion of damage and to show an improvement over the unweighted count of missing embedded resources.

Second, we use our algorithm to assess the damage of mementos in the Internet Archive and WebCite. We compare the M_m and D_m based on Web user agreement with the metrics.

Third and finally, we measure damage in the Internet Archive and WebCite over time using D_m. We describe how this algorithm can be used for future enhancements of the Heritrix crawler [26,37] and Internet Archive's archival processes. We also discuss the impacts of JavaScript on archive quality, using WebCite as the target of our discussion, and compare WebCite's memento quality to Archive.today.

2 Motivating examples

We use the XKCD Web page as an example of a resource with embedded resources of differing importance. We captured the URI-R using the wget [18] command[2] and manually inflicted damage on a local memento of http://www.xkcd.com/ by removing embedded images. We used PhantomJS [30] to dereference the URI-M and take a PNG snapshot of the representation, and we recorded the resulting HTTP response headers of the embedded resources. We created three mementos of the URI-R: One duplicating its live Web counterpart (m_0), one with the central comic image removed (m_1), and one with two logo images removed (m_2). The snapshots taken by PhantomJS are provided in Fig. 1a, b, c. As shown in the captions, the proportion of embedded missing resources to all requested resources (M_m) varies between the mementos.

At the time of this test, the live XKCD site was missing two embedded style sheets, as are m_0, m_1, and m_2 since they are copies of the live site. We verified that our memento m_0 has a M_m value identical to its live Web counterpart—the live resource and m_0 are both missing the same embedded resources ($M_m = 0.17$). In Fig. 1a, m_0 has multiple embedded resources, but we focus on the three identified by the red arrows: the XKCD logo, the main comic image, and the banner of comics. The central image is most important to the utility of the page—without the main comic image, the user does not obtain the information from the page that the author intended (Fig. 1b). The logo and banner are not essential to the user's understanding of the XKCD content (Fig. 1c).

Cascading Style Sheets (CSS) also differ in importance. Some style sheets are responsible for formatting small portions of a page, while others are responsible for placing images and other content or even organizing the entire page for the user. Figure 1d shows a memento of a URI-R that is missing a single style sheet. This style sheet is responsible for a large amount of information in the representation, and

[1] According to the text at https://archive.org/web/ at the time of authoring.

[2] We executed the wget command with parameters as follows: `wget -E -H -k -K -p` http://www.xkcd.com/.

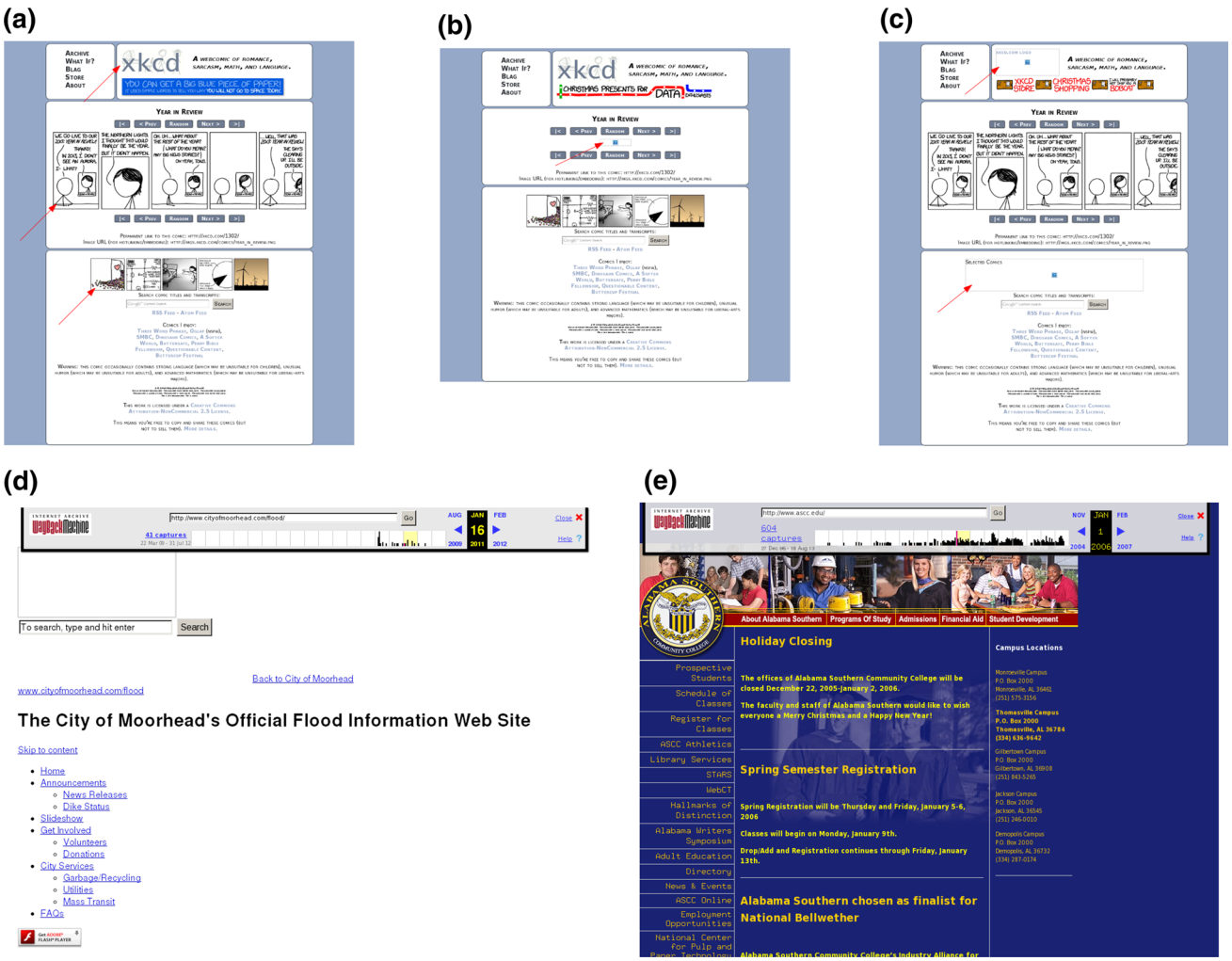

Fig. 1 Mementos have different meanings and usefulness depending on which embedded resources are missing from the memento (and the proportion of missing resources, M_m). **a** All three of the embedded images are included in m_0 and identified by the *red arrows* ($M_m = 0.17$). **b** We removed the large, central image (that is the main content of the page) from m_1, identified by the *red arrow* ($M_m = 0.24$). **c** We removed the XKCD logo and banner of comics from m_2, identified by the *red arrows* ($M_m = 0.29$). **d** This memento (URI-M http://web.archive.org/web/20110116022653/ http://www.cityofmoorhead.com/flood/?) is missing a single style sheet which changes the entire appearance and utility of the memento ($M_m = 0.38$). **e** Meanwhile, this memento (URI-M http://web.archive.org/web/20060102083228/ http://www.ascc.edu/) is missing two style sheets (along with two images) but does not appear damaged ($M_m = 0.20$) (color figure online)

without it, the meaning and utility of the memento change. Figure 1e shows a memento that is properly styled but is missing two style sheets that are not responsible for the majority of the content organization.

As we have discussed, the percentage of successfully dereferenced embedded resources is not the only factor in determining memento quality. In support of that principle, we refer to Fig. 1e in which $M_m = 0.2$ (6/30). However, it appears to be well preserved. In our XKCD example, Fig. 1c is missing two images ($M_m = 0.29$) yet maintains more important embedded mementos than Fig. 1b ($M_m = 0.24$). These examples support the motivation of our research by demonstrating that unweighted percentages (i.e., M_m) are insufficient to assess perceived memento damage.

3 Related work

Researchers have studied the completeness of the archives, the recrawl policies that optimize archive quality, and the relative importance of content within Web resources. We build upon these prior works and apply their findings to develop our algorithm for automatically assessing the quality of mementos.

SalahEldeen et al. have studied the rate at which live resources disappear from the Web. In a study of the Egyptian Revolution, SalahEldeen found that 11 % of the resources shared over Twitter were missing after 1 year [34,36].

Our previous work studied the factors influencing archivability, including accessibility standards and their impact

on memento completeness [22]. In this work, we used a yearly sampling method to select mementos for testing. We use a similar method in this work to study memento damage.

Spaniol et al. measured the quality of Web archives based on matching crawler strategies with resource change rates and related implications for crawling strategies [14,40,41]. Ben Saad and Gançarski performed a similar study regarding the importance of changes on a page [7–9]. Gray and Martin created a framework for high-quality mementos and assessing their quality by measuring the missing embedded resources [19]. While these studies focused on memento completeness and site coverage, we focus on assessing the importance of the artifacts that are missing.

Banos et al. [6] created the CLEAR algorithm to evaluate archival success based on adherence to standards for the purpose of assigning a resource archivability score. The authors expanded on CLEAR and created CLEAR+ in their follow-on efforts [5].

Fersini et al. [17] studied the importance of information blocks of a rendered Web page, finding that blocks with more images are more important. Singh et al. [38] found that multimedia within a page is essential for user understanding. Ye et al. found that the information blocks close to the center of the viewport contain important information, while "noise"—or unimportant content—occurs on the fringes or edges of the page [45]. Kohlschütter et al. [24] also found that important content was located in the center of pages. Centrality is a way for authors to convey importance of information to their users. For example, images in the center of the viewport are more important or contribute more to the users' understanding of a page than those positions on the fringes or outside the viewport of a page. Using these prior findings, we constructed an algorithm to assess the importance of embedded resources based on their MIME type, location in the viewport, and size in pixels.

Zhang et al. [46] studied human perception and human ability to recognize differences in images effectively determining human perception limitations for images at the pixel level. Rademacher et al. [31] used human perception to identify the visual factors that distinguished computer-generated images from photographs. We use human perception in a similar way to identify levels of memento damage.

The algorithm proposed in this paper determines the importance of embedded resources. Song et al. [39] outlined an algorithm for determining the importance of sections of Web pages based on their content, size, and position. Song's work focused on recognizing important blocks of a Web page to eliminate noise in an effort to accurately extract aspects of pages that users would find most important. Blocks featured prominently in the center of the view port and occupying a large area of the page were found to be most important. We utilize this concept, identifying content occupying large amounts of viewport real estate as important in our measurements of the importance of missing embedded resources.

4 Users' perception of damage

As archivists, our perception of damage differs from that of more traditional Web users. To determine whether M_m (percent missing) is a good estimate of human perception of damage, we used Amazon's Mechanical Turk to measure human agreement with M_m.

To ensure that Mechanical Turk workers (or more colloquially, "turkers") could evaluate damage, we presented turkers with pairs of mementos that had varying levels of damage and asked them to select the memento they preferred to keep if given a choice between the two.

We captured 11 hand-selected URI-Rs (Table 1) on a local server and created five versions of the mementos for each URI-R. We manually damaged the mementos to create the five categories of damage. For the category *missing image*, we removed a prominent image (empirically identified as important) from the memento. For the category *missing css*, we removed a prominent CSS file to cause formatting issues in the memento; we empirically selected the CSS file to remove based on the greatest human-perceived detrimental impact to the page layout. We also created the categories *missing all images* (we removed every embedded image), *missing all resources* (we removed all embedded resources), and *original* (the URI-M was a direct copy of the live resource) and measured the M_m of each URI-M in each category. We refer to the four categories of damaged mementos in aggregate as m_1 and the *original* as m_0. These categories created several degrees of damage through a variety of missing embedded resources for identical URI-Rs at an identical time point to provide a wide spectrum of mementos to be evaluated by turkers.

With the goal of determining whether or not turkers can recognize damage in a memento, we presented the turkers with an m_1 and its m_0 counterpart (that is, a "damaged" and its *ground-truth* memento) and asked the turkers "We saved two versions of the same website ... *Which version did we do a better job saving?*" (Fig. 2). For each URI-R, a pair of mementos consisting of m_0 and one of the four categories of m_1 were evaluated by five turkers for a total of 280 evaluations. We follow the precedent of using five turkers to establish turker opinion as established by SalahEldeen and Nelson [35].

We show the judgement splits from the turker evaluations in Table 2. The judgement splits refer to the number of turkers that selected the correct–incorrect version. For example, a 0–5 split means all five turkers selected the m_1 (an incorrect selection), a 5–0 split means all five turkers selected the m_0 memento (the correct selection), and a 3–2 split means three

Table 1 The 11 URI-Rs used to create the manually damaged dataset. M_m values are provided for each m_1

URI-R	M_m				
	m_0	Missing image	Missing css	Missing all images	Missing all
http://www.cs.odu.edu/~mln/	0.14	0.43	0.29	0.43	0.43
http://activehistory.ca/2013/06/myspace-is-cool-again-too-bad-they-destroyed-history-along-the-way/comment-page-1/	0.0	0.32	0.32	0.57	0.85
http://www.albop.com/	0.0	0.13	0.0	0.50	0.50
http://www.cs.odu.edu/	0.10	0.13	0.11	0.82	0.81
http://ws-dl.blogspot.com/2013/08/2013-07-26-web-archiving-and-digital.html	0.07	0.08	0.08	0.13	0.14
http://www.cnn.com/2013/08/19/tech/social-media/zuckerberg-facebook-hack/	0.19	0.22	0.28	0.46	0.57
http://xkcd.com/	0.14	0.38	0.31	0.53	0.54
http://www.mozilla.org/	0.80	0.80	0.80	0.877	0.89
http://www.ehow.com/	0.05	0.05	0.06	0.11	0.33
http://google.com/	0.0	0.0	0.0	0.0	1.0
http://php.net/	0.32	0.33	0.33	0.37	0.37

turkers selected the m_0 memento and two selected the m_1 (a correct selection by the majority, but still a split decision among the turkers). For the purposes of this paper, we consider only 5–0 and 4–1 splits as agreement with M_m and all other splits as disagreement. ΔM_m refers to the delta between M_{m_0} and M_{m_1}.

The turkers selected m_0 as the preferred option (less damaged memento) 81 % of the time (226/280). As ΔM_m grows, turker agreement is more consistent.

Regardless of ΔM_m, 81 % of the evaluations agreed with M_m as a suitable damage metric (5–0 and 4–1 splits). Turkers were unsure about the damage (3–2 and 2–3 splits) 18 % of the time and incorrectly identified damage only once. The average ΔM_m for the unsure selections was < 0.01, and the only 0–5 split had a ΔM_m of 0.014, suggesting that confusion or disagreement occurs more often when the damage delta is smaller.

Confusion matrices provide a consolidated view of an algorithm's performance. The top-left quadrant shows the number of true positives, the top right shows the number of false negatives, the bottom left shows false positives, and the bottom right shows true negatives. The algorithm's accuracy ((True Positives + True Negatives) / (All Positives and Negatives)) and harmonic mean (or F_1 Score: 2 × True Positives / (2 × True Positives + False Positives + False Negatives)) are calculated using a confusion matrix. A harmonic mean provides an average (in this case, of the algorithm's success rate) and is sensitive to small values and outliers.

From the confusion matrix (Table 3), we can calculate $F_1 = 0.88$ and accuracy=0.80 for m_0 versus m_1. Turker agreement does not match M_m 100 % of the time with the m_0 versus m_1 test because of phenomena with esthetics and human perception. Also, m_0 is often incomplete ($M_{m_0} > 0$) and, as a result, has $D_{m_0} > 0$[3] (see php.net in Table 1).

Ideally, turker agreement would be unanimous. The measured turker agreement of 81 % can be attributed to one of several factors. The ΔM_m measures were very small in comparisons resulting in split turker decisions, potentially causing the damaged mementos to look very similar based on human perception. This phenomenon further illustrates the need for a D_m measure because turkers have dissenting opinions when mementos are damaged in visually similar ways. Regardless of the reason the turker agreement fell short of 100 % with M_m, we demonstrate an improvement of D_m over M_m.

5 Evaluating organic damage

Having identified m_0 in the m_0 versus m_1 in a large majority (81 %) of the comparisons, the turkers have shown that they can identify a damaged resource when presented a damaged and undamaged memento. Because they can identify damage in mementos, we used turkers to evaluate our measured damage of mementos found in the Internet Archive.

5.1 Dataset selection

This experiment uses the same set of 2,000 URI-Rs as in our previous work [12], which was sampled from Twitter and Archive-It. The first dataset, the *Twitter* set, consists of 1,000

[3] Live Web resources may have missing embedded resources, and this results in a calculated $D_{m_0} > 0$.

Fig. 2 We asked the turkers to select the less damaged of two mementos. The two versions of the page are accessible in separate tabs

Table 2 The turkers selected m_0 as the preferred memento 81 % of the time, and more consistently for larger ΔM_m values

ΔM_m	Splits						
	5–0	4–1	3–2	2–3	1–4	0–5	Total
1.0							0.00
0.9							0.00
0.8	4						0.07
0.7							0.00
0.6							0.00
0.5	1	1					0.04
0.4							0.00
0.3	15	5					0.36
0.2	2						0.04
0.1	5	4	4	2		1	0.29
0.0	5	3	1	3			0.22
Total	0.58	0.23	0.09	0.09	0.00	0.02	1.0

Table 3 Confusion matrix of the turker assessments of the m_0 versus m_1 comparison test

Turker assessment	M_m	
	Select m_0	Select m_1
m_0	44	0
m_1	11	0

Bitly URIs shared over Twitter and represents a more random selection of URI-Rs not explicitly selected for curation by human archivists. We collected the Twitter URIs through the Twitter Garden Hose[4] in October 2012.

The second dataset, the *Archive-It* set, was sampled from Archive-It collections. Archive-It collections are created and curated by human archivists often corresponding to a certain event (e.g., National September 11 Memorial Museum) or a specific set of Web sites (e.g., City of San Francisco). The Archive-It set consists of the entire set of URI-Rs belonging to the collections listed on the first page of collections at Archive-It.org[5] as of October 2012. This resulted in 2,093 URI-Rs that represent a collection of previously archived and human-curated URIs. To make the datasets equal in size, we randomly sampled 1,000 URI-Rs from the set of 2,093.

We discarded non-HTML representations (e.g., JPEG and PDF) from both sets and combined the Twitter and Archive-It datasets for a final dataset of 1861 URI-Rs. Non-HTML representations do not contribute to this study since they do not have embedded resources. There is no overlap between the two sets.

As measured in our prior work [12], the resources in the Archive-It set receive an HTTP 200 response for 93.5 % of all requests for embedded resources and the resources in the Twitter set receive an HTTP 200 response for 87.1 % of all requests for embedded resources.

5.2 Turker evaluation

Using this set of URI-Rs, we measured the damage of one memento per year from the Internet Archive TimeMaps of each of the 1,861 URI-Rs, resulting in 45,341 URI-Ms. We randomly selected a subset of 100 URI-Ms from this set. Similar to the evaluation in Sect. 4, we gave turkers two mementos (we will generalize these to m_2 and m_3) from consecutive years from the same TimeMap and asked the turkers to select the less damaged memento ("We saved two versions of the same website ... *Which version did we do a better job saving?*") as shown in Fig. 2. Because m_2 and m_3 are observed from the Internet Archive, neither is considered a *ground-truth*. We measured M_m of mementos in the Internet

[4] https://dev.twitter.com/docs/streaming-apis/streams/public.

[5] http://www.archive-it.org/explore/?show=Collections.

Table 4 The turker evaluations of the m_2 versus m_3 comparisons when using M_m as a damage measurement

ΔM_m	Splits						
	5–0	4–1	3–2	2–3	1–4	0–5	Total
1.0					1		0.01
0.9							0.00
0.8							0.00
0.7		1					0.01
0.6					1		0.01
0.5							0.00
0.4		1					0.01
0.3	1		3	4	1	2	0.11
0.2		5	6	5	12	9	0.37
0.1	4	5	10	11	15	3	0.48
0.0							0.00
Total	0.05	0.12	0.19	0.20	0.30	0.14	1.0

Table 5 Confusion matrix of the turker assessments of the m_2 versus m_3 comparison test against M_m

Turker	M_m	
Assessment	Select m_2	Select m_3
m_2	29	24
m_3	23	24

Archive and compared it to the turker perception of the utility of the mementos.

Contrary to the test in Sect. 4, as ΔM_m grows, the turkers are not as effective at selecting the less damaged memento (the splits are shown in Table 4). The turkers only agree with M_m 12 % of the time and completely disagree with M_m (1–4 and 0–5 splits) 44 % of the time. This discrepancy demonstrates that turker assessment of damage does not match M_m. Additionally, we see that the turkers performed well when comparing m_0 versus m_1 (original versus damaged) but struggle to compare m_2 versus m_3 (damaged vs damaged).

From the confusion matrix (Table 5), we can calculate the accuracy of turker selections of m_2 versus m_3 agreement with M_m is 0.46 with $F_1 = 0.55$. In a receiver operating characteristic (ROC) curve [16], we calculated the area under the ROC curve (AUC) for the results of the turker evaluations of m_2 versus m_3 against M_m and the results of the manually damaged m_0 versus m_1 test. The AUC of M_m is lower (AUC = 0.472) than random (AUC = 0.500) as shown in Table 6, meaning that M_m performed worse than random for matching turker perception of damage.

6 Calculating memento damage

With M_m not matching Web users' perception of damage, we propose a new algorithm for assessing memento damage.

Table 6 When compared to random, M_m performs worse than random selection and is worse than the performance of m_0 versus m_1

Damage calculation	AUC	F_1	Accuracy
M_m	0.472	0.55	0.46
M_{m_0}	0.789	0.88	0.80

Our proposed algorithm is based on the MIME type, size, and location of the embedded resource.

6.1 Defining D_m and M_m

Before defining equations for our memento quality measurements, we first describe the resources in the mementos in Eq. 1, differentiating between the set of all embedded resources R and the set of all missing resources R_m. In this case, we consider any resource needed to build a resource and that is requested by the client an *embedded resource*.

$$\begin{aligned} R &= \{\text{All embedded resources requested}\} \\ R_m &= \{\text{All missing embedded resources}\} \\ R_m &\subseteq R \end{aligned} \tag{1}$$

As we mention in Sect. 2, we calculate R by counting the number of distinct and unique URIs requested by the client when dereferencing the URI-M. For example, if an image identified by $URI - R_a$ is referenced three times in the DOM, it is only requested once by the client and is only counted once in R. Similarly, we calculate R_m by counting only the URI-Rs that, when dereferenced, return an HTTP response code in the 400 or 500 range (i.e., is not successfully dereferenced). If an HTTP GET for $URI - R_a$ returns an HTTP 404 response (or an HTTP 503 response), it counts once in R_m.

Our measurement of M_m is the proportion of missing embedded resources to all requested resources (Eq. 2). We define M_m as a proportion because it normalizes the measurement. Without using a proportion, M_m breaks down when mementos have a very large or very small number of embedded resources. For example, a memento with two embedded resources and is missing one of the two embedded resources has a lower archiving success rate than a memento with one hundred embedded resources and is missing one of the one hundred embedded resources. Normalizing M_m allows use to compare mementos that have different numbers of embedded resources using the same metric.

The M_m measure includes resources that were omitted from a crawl due to crawl policies or robots.txt [42] because the goal of M_m and D_m is to help identify damage independently of conscious efforts of the archival institutions.

$$M_m = \frac{R_m}{R} \tag{2}$$

We define D_m as the damage rating, or cumulative damage, of a memento m in Eq. 3. D_m is a normalized value ranging from [0, 1]. We calculate the potential damage of a memento and the actual damage of a memento and express the damage rating as the ratio of actual to potential damage. Notionally, potential damage is the cumulative importance of all embedded resources in the memento, while actual damage is only the importance of those embedded resources that are unsuccessfully dereferenced, or missing.

$$D_m = \frac{D_{m_{\text{actual}}}}{D_{m_{\text{potential}}}} \tag{3}$$

6.2 Weighting embedded resources

We calculate the importance of each embedded resource in the set R. The sum of each embedded resource is the potential damage $D_{m_{\text{potential}}}$ (Equ. 4). Important resources are assigned additional weights to increase their relative value over unimportant resources (Eqs. 6–7).

$$\begin{aligned} D_{m_{potential}} &= \frac{\sum_{i=1}^{n_{[I,MM]}} D_{[I|\text{MM}]}(i)}{n_{[I|\text{MM}]}} + \frac{\sum_{i=1}^{n_C} D_C(i)}{n_C} \\ &\forall\{\text{I} = \text{Images}, \text{MM} = \text{Multimedia}, \text{C} = \text{CSS}\} \\ &n \in R \end{aligned} \tag{4}$$

Actual damage ($D_{m_{\text{actual}}}$, defined in Eq. 5) is identical to $D_{m_{potential}}$ except it is computed using only the missing embedded resource set R_m.

$$\begin{aligned} D_{m_{\text{actual}}} &= \frac{\sum_{i=1}^{n_{[I,\text{MM}]}} D_{[I,\text{MM}]}(i)}{n_{[I|\text{MM}]}} + \frac{\sum_{i=1}^{n_C} D_C(i)}{n_C} \\ &\forall\{\text{I} = \text{Images}, \text{MM} = \text{Multimedia}, \text{C} = \text{CSS}\} \\ &n \in R_m \end{aligned} \tag{5}$$

In M_m, as opposed to D_m, all embedded resources are considered equal. The potential damage is therefore the number of embedded resources, and the actual damage is the number of missing embedded resources. M_m is the unweighted ratio of missing embedded resources to total embedded resources.

We introduce additional weights of differing values to account for the notion of embedded resource importance. When a weight w is given to an embedded resource, all n embedded resources lose $\frac{w}{n}$ importance, which redistributes the importance between embedded resources while keeping the sum of all importance constant. Note that we only assign additional weights to embedded resources that are visually validated as present (i.e., images, multimedia, and style sheets); the weighted importance of other embedded resources is considered outside of the scope of this research.

6.2.1 Image damage calculation

To account for image importance, images receive weights w for image size and centrality (Eq. 6). We use the pixel area (width x height) of the image as specified in the HTML and the page size along with a weight for horizontal and vertical central dividing line overlap by the image. We omit the size attribute from the calculation if the image dimensions are missing from the HTML. For example, we can extract the width and height of the missing embedded resource "IMAGE" from this HTML

```
<img src=''IMAGE.png'' height=''42'' width=''42''>
```

but not this HTML

```
<img src=''IMAGE.png''>.
```

$$\begin{aligned} D_{[I|\text{MM}]} &= 1 + \frac{\text{width} \times \text{height}}{\text{Page Size (pixels)}} + w_{\text{horizontal}} + w_{\text{vertical}} \\ w_{\text{horizontal}} &= \begin{cases} 0.25 & \text{image overlaps horizontal center} \\ 0 & \text{otherwise} \end{cases} \\ w_{\text{vertical}} &= \begin{cases} 0.25 & \text{image overlaps vertical center} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

Embedded multimedia importance (D_{MM}) is calculated identically to image importance D_I, and we represent both in the same equation $D_{[I|MM]}$. Because size and centrality determine multimedia importance, we omit audio and other non-visual multimedia resources. We also classify Flash movies as multimedia.

6.2.2 Style sheet damage calculation

Equation 7 outlines the damage from missing style sheets, including a factor for a style threshold w_{style} and a threshold for non-matching CSS tags in the DOM w_{tags}.

$$\begin{aligned} D_C &= 1 + w_{\text{style}} + w_{\text{tags}} \\ w_{\text{style}} &= \begin{cases} 0.50 & > 75\,\% \text{ content in left two-thirds} \\ 0 & \text{otherwise} \end{cases} \\ w_{\text{tags}} &= \begin{cases} 0.50 & \text{tags in DOM but not CSS} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

Traditional Web design (and particularly design enabled by style sheets) evenly distributes content across each of the vertical thirds of a page. Our intuition is that a missing important style sheet will shift content to the left of the page rather than center content in the viewport. To identify this phenomenon, we divide the PNG snapshot of a memento into vertical thirds and measure the amount of content in each third. If a

style sheet is missing and content appears to be shifted to primarily the left two-thirds, we assume the missing style sheet was important to the distribution of content on the page.

When detecting content in the PNG snapshot, we use remaining CSS files and the HTML to determine the background color of the page. We measure the number of background- and non-background-colored pixels, with content being the number of non-background-colored pixels. The proportion of non-background-colored pixels in each vertical third gives us the amount of content in each partition.

The style threshold is determined as follows:

1. Determine background color
2. Render a PNG snapshot of the page
3. Divide PNG into vertical third partitions
4. Calculate number of pixels of the non-background color in each third for the viewport only (we used a 1024x768 viewport) and entire page
5. If $\leq$75 % of the non-background-colored pixels are in the left two-thirds of the viewport, set $w_{\text{style}} = 0$ in Eq. 7 (CSS file does not receive a weight)
6. If >75 % of the non-background-colored pixels are in the left two-thirds of the viewport and left two-thirds of the entire page and a style sheet is missing, $w_{\text{style}} = 0.5$ in Eq. 7 (CSS file does receive a weight)

For example, we created two mementos of the URI-R http://www.pilotonline.com/ on a local server, one as it appears live (with all style sheets—Fig. 3a) and the other with its style sheets removed (Fig. 3b). The vertical partitions extend from the top of the PNG snapshot to the bottom. The percent of non-background color pixels in the viewports of our mementos are shown in their respective thirds in Fig. 3a, b. Notice that the non-background pixels (text, images, etc.) shift left when the CSS is missing. Intuitively, information is not meant to be displayed like the content in Fig. 3b.

When we consider content outside of the viewport (Fig. 4a, b), we see the same shift of content to the left when style sheets are missing. However, the distribution of content in Fig. 4b is more evenly distributed because the content has shifted down and fills out the middle and right vertical partitions more than in Fig. 3b. This is an indicator that the style sheets missing in Figs 3b and 4b were important.

Along with the style threshold, the presence of tags on the page without a matching style suggests that the missing CSS contained the referenced formatting. If such tags exist without a matching style, $w_{\text{tags}} = 0.5$ in Eq. 7.

6.3 The D_m Algorithm

Embedded multimedia, images, and style sheets do not account for the entirety of a page's importance and usefulness. We assume that text, as defined by the DOM and included on the page, is available regardless of archival success and therefore does not contribute to the damage calculation.

Fig. 3 Missing style sheets cause content to shift left. We show the percent of content in the vertical partitions of the viewport. **a** We calculated that the *non-background color* is more evenly distributed between the three vertical partitions of the Pilot Online page with its style sheet included than when it is missing. **b** We calculated that the *non-background color* is most prevalent in the left-most vertical partition of the viewport of the Pilot Online page when it is missing its style sheet (color figure online)

In summary, Eqs. 3–7 are used to compute D_m in the following manner:

1. Load URI-M with PhantomJS
2. Find potential damage $D_{m_{\text{potential}}}$ (Eq. 4)
   (a) Determine CSS importance D_C (Eq. 7)
   (b) Determine multimedia importance D_{MM} (Eq. 6)
   (c) Determine image importance D_I (Eq. 6)
3. Determine proportion of unsuccessfully dereferenced embedded resources M_m (Eq. 1)
4. Find actual damage $D_{m_{\text{actual}}}$ (same as Step 3, but with only those URI-Ms unsuccessfully dereferenced R_m)
5. Determine total damage $D_m = [0,1]$ (Eq. 5)

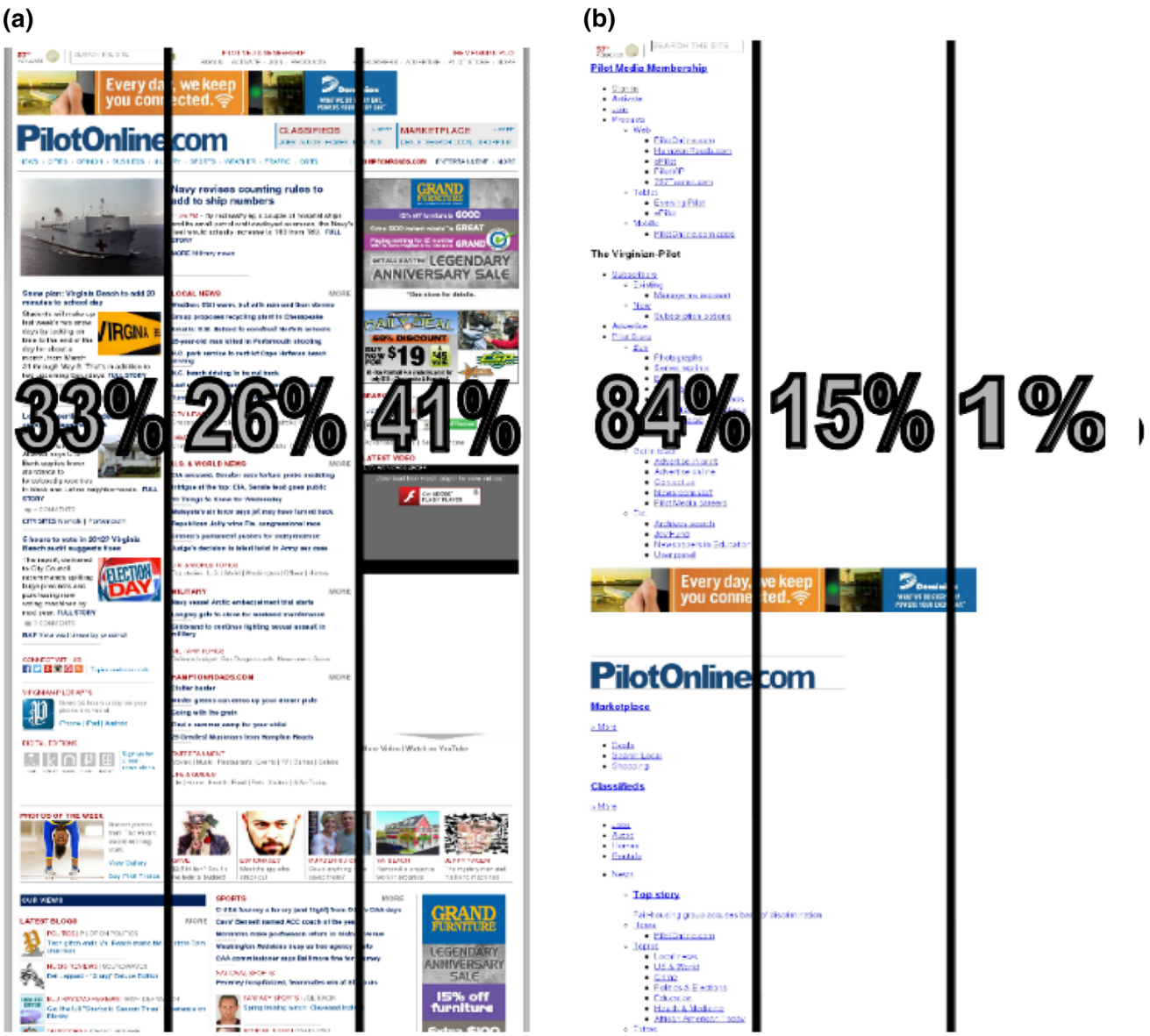

Fig. 4 The left shift caused by a missing style sheet occurs throughout the entire page and is not limited to the viewport. **a** When considering the entire page, the content of the page is distributed 33 % in the left, 26 % in the middle, and 41 % in the right partitions when the style sheet is present. **b** When considering the entire page, the content of the page is distributed 84 % in the *left*, 15 % in the *middle*, and 1 % in the right partitions when the style sheet is missing

With D_m defined, we revisit the examples presented in Sec. 2. The values for D_m and M_m are listed in Table 7. Note that the D_m ratings are closer to our empirical human assessment of memento quality than the proportion of the embedded resources that are missing.

Table 7 D_m versus M_m for the images in Fig. 1

Figure	D_m	M_m
1a	0.09	0.17
1b	0.41	0.24
1c	0.36	0.29
1d	0.59	0.38
1e	0.003	0.20

$M_m > D_m$ in 2 of 5 cases

6.4 Limitations of D_m calculation

Not all pages and page construction methods can be evaluated by this algorithm. An edge case not handled by this algorithm is any page constructed with iframes. Our algorithm uses JavaScript to determine the rendered location of embedded multimedia and images. When the embedded media are in a

page embedded within another page, our algorithm does not provide the accurate rendered location. For this reason, we exclude iframes from our algorithm. We also exclude missing audio-only multimedia.

While D_m includes multimedia calculations, multimedia resources are rarely embedded in our mementos (only observed twice in our entire set of 45,341 URI-Ms). We observed that multimedia is often loaded by JavaScript files embedded in the document object model (DOM); this prevents the multimedia files from being archived since archival crawlers (at the time of this experiment) do not execute client-side JavaScript and therefore do not discover the requested files.

Further, the JavaScript files may not operate properly when archived [11] and may not issue a request for the target multimedia files. If the JavaScript operates properly and makes an HTTP GET request, the multimedia file would be missing (since it is not archived) and we would observe more missing embedded multimedia files. We discuss this issue further in Sect. 8.1.

The D_m measurement and its constituent weights were constructed by archivists as an improvement to the metric M_m currently used for archive quality assurance. We do not assert that D_m is a perfect measure, but rather an improvement that will require additional investigation and re-weighting to reach perfect agreement with turker evaluation. We recognize that D_m should be more finely tuned to more accurately reflect turker opinion of damage. We also avoid defining a threshold for damage acceptance; this is left to the discretion of the archivist utilizing D_m to measure damage in an archive.

6.5 Turker assessment of D_m

We compared D_m to turker assessment and to M_m. As shown in Table 8, D_m agrees with turker assessment of damage 32 % of the time, an increase of 18 % over M_m. Additionally, 49 % tie with a 3–2 or 2–3 split and only 16 % of the turker evaluations disagreed with the D_m measure. Turkers agree more consistently when ΔD_m is larger. If we only consider $\Delta D_m > 0.30$, the turkers agree with D_m 71 % of the time. However with $\Delta M_m > 0.30$, the turkers agree only 20 % of the time.

From the confusion matrix in Table 9, we determine that the accuracy of D_m when comparing m_2 versus m_3 is 0.60, and $F_1 = 0.69$. This is an improvement of 0.14 over the accuracy of M_m and an improvement over the harmonic mean of M_m by 0.14, showing that D_m measures damage closer to turker perception. We also calculated the AUC in a ROC curve for D_m and compared it to M_m and the performance of the m_0 versus m_1 test. As shown in Table 10, D_m has an AUC of 0.584, an increase in 0.108 over M_m, showing that D_m outperforms M_m and is closer to the performance of m_0 versus m_1 (AUC = 0.789).

Table 8 The turker evaluations of the m_2 versus m_3 comparisons when using D_m as a damage measurement

ΔD_m	Splits						
	5–0	4–1	3–2	2–3	1–4	0–5	Total
1.0							0.00
0.9		1					0.01
0.8							0.00
0.7							0.00
0.6			1				0.01
0.5							0.00
0.4	4	1					0.05
0.3	2	2	3				0.07
0.2		2	1	2	2	1	0.08
0.1	4	16	27	15	12	3	0.77
0.0							0.00
Total	0.10	0.22	0.32	0.17	0.14	0.04	1.0

Table 9 Confusion matrix of the turker assessments of the m_2 versus m_3 comparison test against D_m

Turker assessment	D_m	
	Select m_2	Select m_3
m_2	45	32
m_3	8	14

Table 10 D_m provides a closer estimate of turker perception of damage and our performance of m_0 versus m_1 than M_m

Damage calculation	AUC	F_1	Accuracy
M_m	0.472	0.55	0.46
D_m	0.584	0.69	0.60
M_{m_0}	0.789	0.88	0.80

7 Damage in the archives

Having defined an algorithm for measuring D_m, we measured D_m values for each of the 45,341 URI-Ms from Sect. 5. We used these measurements to assess D_m's performance relative to turker assessment and to perform damage measurements in the Internet Archive.

7.1 Measuring the Internet Archive

With D_m validated as aligning closer to turker evaluations than M_m, we used D_m to evaluate the Internet Archive's performance. Our measurement shows that only 46 % of the

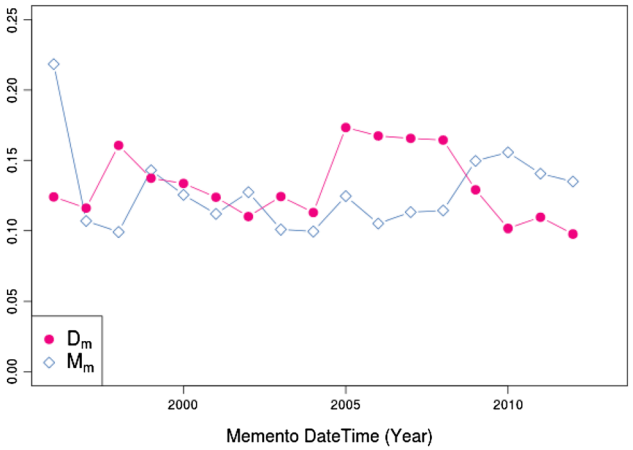

Fig. 5 The average embedded resources missed per memento per year in the Internet Archive as compared to damage per memento per year ($\overline{D_m} = 0.128$, $\overline{M_m} = 0.132$)

45,341 URI-Ms listed in the 1,861 TimeMaps are complete—that is, 54 % of all URI-Ms listed in the Internet Archive TimeMaps we studied are missing at least one embedded resource.[6] In Fig. 5, we show the average number of missing embedded resources M_m along with the average calculated damage D_m per URI-M per year.

Because the number of missed mementos is important to M_m and D_m, we investigated the occurrence of missing and successfully dereferenced embedded resources. Most mementos are missing very few embedded resources with most missing 1–10 embedded resources (as a histogram and cumulative distribution function (CDF) in Figs. 6a, b), ($\mu = 1.7, \sigma = 4.6, \text{median} = 3$). We note the long tail on this distribution; a few mementos are missing a larger amount of embedded resources (maximum is 116). We calculate that 61 % of mementos are missing 3 or fewer embedded resources, and 85 % of mementos are missing 6 or fewer embedded resources. Most mementos have very few embedded resources ($\mu = 17.6, \sigma = 86, median = 7$), as shown in Fig. 6c, d). A few mementos have a very large number of embedded resources (maximum is 552).

In aggregate, we observed that 45,009 of 292,192 embedded resources were missing, meaning 15 % of the embedded resources in the dataset are missing. Of these, 25,848 (57 % of the missing URI-Ms) were important, meaning they were assigned an additional weight by D_m (Eqs. 5, 6). The average damage of all measured mementos was 0.132.

The yearly $\overline{D_m}$ goes from 0.16 in 1998 to 0.13 in 2013. That means the Internet Archive is doing a better job (over time) reducing the total memento damage in its collection. However, the number of missing *important* resources (resources with an importance > 1 due to added weights) is increasing, going from an average of 1.30 important resources per memento in 1997 to 2.38 important resources per memento in 2013 for an average of 2.05 missing per memento. Meanwhile, the number of unimportant missing embedded resources (damage rating weight ≤ 1) per memento is increasing at a lesser rate, going from 1.35 in 1997 to 1.64 in 2013. This suggests that while the Internet Archive is getting better overall at mitigating damage as much as possible, the archive is missing an increasing number of embedded resources deemed important.

The distribution of file types missing per memento (Fig. 7) shows that most URI-Ms are missing ≥ 1 embedded resource and that style sheets and JavaScript files are missing at higher rates over time. Missing JavaScript may lead to additional missing files (such as multimedia). Images are missing at varying rates per memento over time.

7.2 Measuring WebCite

In an effort to measure a less prominent and different type of archive, we used the damage algorithm to determine M_m and D_m of WebCite.[7] WebCite [15] is different from the Internet Archive's Heritrix crawler in that it is a page-at-a-time (i.e., crawls a single URI-R and not an entire site) archiving tool that creates mementos upon user request.

Web crawlers like Heritrix operate by starting with a finite set of seed URI-Rs in a frontier—or list of crawl targets—and add to the frontier by extracting embedded URIs in the representations of the URI-R. This allows archival crawlers to discover embedded resources as well as new URI-Rs to crawl while creating mementos.

The Internet Archive follows this model with the goal to archive the Web using the Heritrix crawler, while WebCite and other page-at-a-time archivers allow users to submit URI-Rs for archiving, and WebCite immediately archives the resource.[8] When using a page-at-a-time archival service, the resulting memento contains embedded resources with the same archival datetime [1]. This section identifies our damage measurement of this page-at-a-time archiver and outlines the differences between Heritrix and WebCite.

Our WebCite dataset has 992 mementos in 285 TimeMaps of our collection of 1,861 URI-Rs. The earliest available memento is from 2007, and the latest is from 2014. Only six mementos are available from 2014; therefore, we will focus on 2007–2013 as the target years of investigation due to the limited number of 2014 mementos, as well as to match the period of observation of the Internet Archive. The $\overline{D_m}$ of

[6] The Internet Archive performs URI canonicalization very well and is assumed to not be a source of missing resources.

[7] http://webcitation.org/.

[8] The Internet Archive has recently added an *on-demand* archiving utility at http://archive.org/web/ under the heading "Save Page Now" [33].

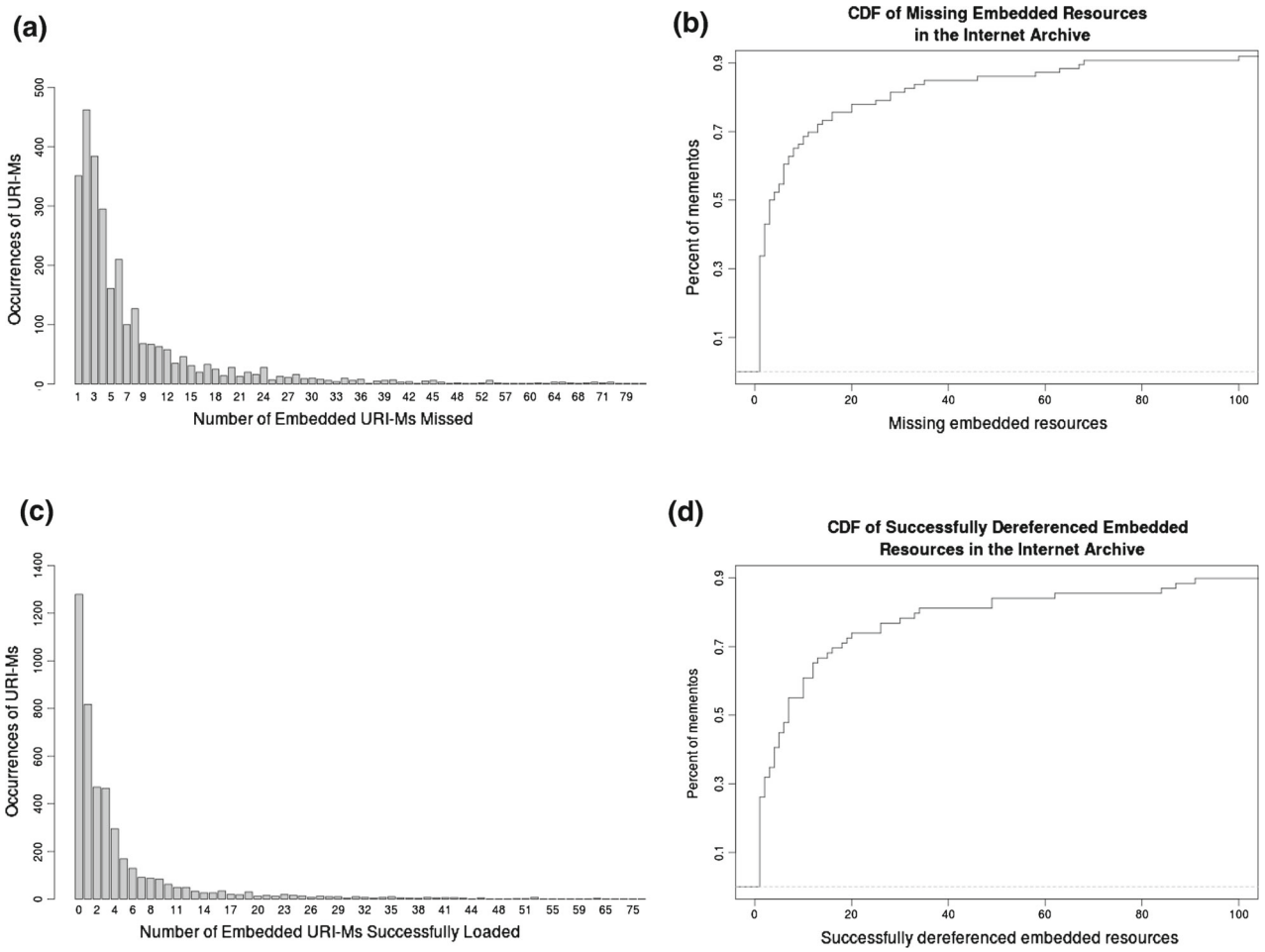

Fig. 6 The distribution of the number of successfully dereferenced and missing embedded resources per URI-M in the Internet Archive. Note that we limited the figures to 100 missing or successfully dereferenced embedded resources, respectively. **a** Occurrences of missing embedded resource numbers in the Internet Archive as a histogram. **b** Distribution of missing embedded resources within the collection of Internet Archive mementos as a CDF. **c** Occurrences of successfully dereferenced embedded resource numbers in the Internet Archive as a histogram. **d** Distribution of successfully dereferenced embedded resources within the collection of Internet Archive mementos as a CDF

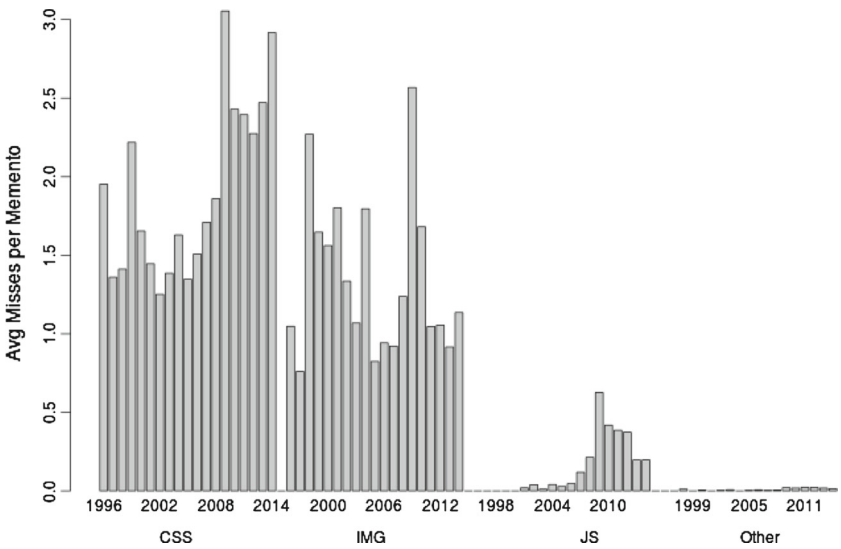

Fig. 7 The number of missed embedded resources per Internet Archive memento per year and MIME type

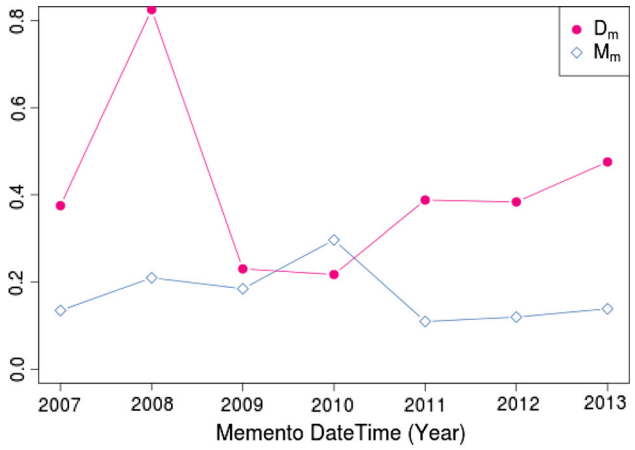

Fig. 8 The average embedded resources missed per memento per year in WebCite as compared to damage per memento per year ($\overline{D_m} = 0.397$, $\overline{M_m} = 0.176$)

the collection over all years is 0.397 ($\sigma = 0.194$), and the $\overline{M_m}$ is 0.176 ($\sigma = 0.0926$). All of the mementos in this collection are missing at least one embedded resource—100 % of the mementos are incomplete.

As shown in Fig. 8, the $\overline{D_m}$ in WebCite is increasing over time, going from 0.285 in 2007 to 0.442 in 2013. Meanwhile, the average M_m remains steady, going from 0.135 in 2007 to 0.139 in 2013. Only slight variation occurs, peaking at 0.287 in 2010.

Compared to the Internet Archive, WebCite has a higher damage value as well as is missing a larger percentage of embedded resources. Additionally, D_m per memento is higher, indicating that a larger percentage of missing embedded resources are important (3,514 or 41.7 %) in WebCite than in the Internet Archive.

WebCite is missing on average 10.1 embedded resources per memento ($\sigma = 8.0$, median $= 2$). This distribution exhibits a long tail, with a few mementos missing a large number of embedded resources (maximum is 133). WebCite mementos successfully dereference on average 15.3 embedded resources per memento ($\sigma = 30.7$, median $= 4$); again note the long tail (maximum is 154). Across the entire collection, 8,420 of 54,824, or 15.4 % of the embedded resources were missing in our investigation. We calculate that 56 % of mementos are missing 3 or fewer embedded resources, and 74 % of mementos are missing 6 or fewer embedded resources (Fig. 9).

The distribution of file types missing per memento (Fig. 10) shows that most URI-Ms are missing ≥ 1 embedded image and CSS resources, on average. WebCite has a lower occurrence of missing style sheets, but a higher occurrence of missing images.

Our previous investigation showed that WebCite has difficulties when encountering JavaScript and embedded iframes [12]. However, its archiving policies provide immediate results as opposed to crawlers that may incur a delay between the time a URI-R is added to the frontier and a memento is created. WebCite's difficulties with JavaScript may contribute to the missing embedded resources if they were loaded through JavaScript.

8 Impact of JavaScript on damage

As a preliminary investigation of the impact of JavaScript on archival tools, we set up an experiment to use Heritrix and PhantomJS to crawl the same set of URI-Rs and measure the damage difference between the two sets of mementos. Our goal is to understand how D_m is impacted by JavaScript by comparing mementos archived by a crawler that can execute JavaScript (PhantomJS) and a crawler that does not execute JavaScript (Heritrix).

8.1 PhantomJS versus Heritrix

Representations of Web resources are increasingly reliant on JavaScript and other client-side technologies to load embedded resources and control the activity on the client or request additional data or resources (e.g., via Ajax) after the initial page load. We refer to representations that are changed by client-side code, such as JavaScript, as *deferred representations* because the full representation is not realized until after the initial page load. Crawlers are unable to discover the resources requested via Ajax and are missing embedded resources which ultimately causes the mementos of the crawled resources with deferred representations to be incomplete and have higher D_m.

To mitigate the impact Web developers' practice of using JavaScript and Ajax to load embedded resources, crawlers like Heritrix have constructed approaches for extracting links from embedded JavaScript to be added to crawl frontiers [21] (most recently, Google's crawler [10]). Archive-It has recently adopted Umbra to archive a hand-selected set of URI-Rs known to have deferred representations [32]. However, this is not a solution to the challenges that JavaScript introduces in the archives, but is a mitigation of the impact for a small set of URI-Rs (e.g., facebook.com, twitter.com URI-Rs).

Because archival crawlers' abilities differ from the abilities of browsers, the archives currently hold a representation of the Web from the point of view of crawlers and not Web users. That is, what we archive is increasingly different than what users experience.

8.2 Crawling deferred representations

We sampled 50 URI-Rs by randomly generating Bitly.com URIs and identifying the URI-Rs to which the Bitly URIs

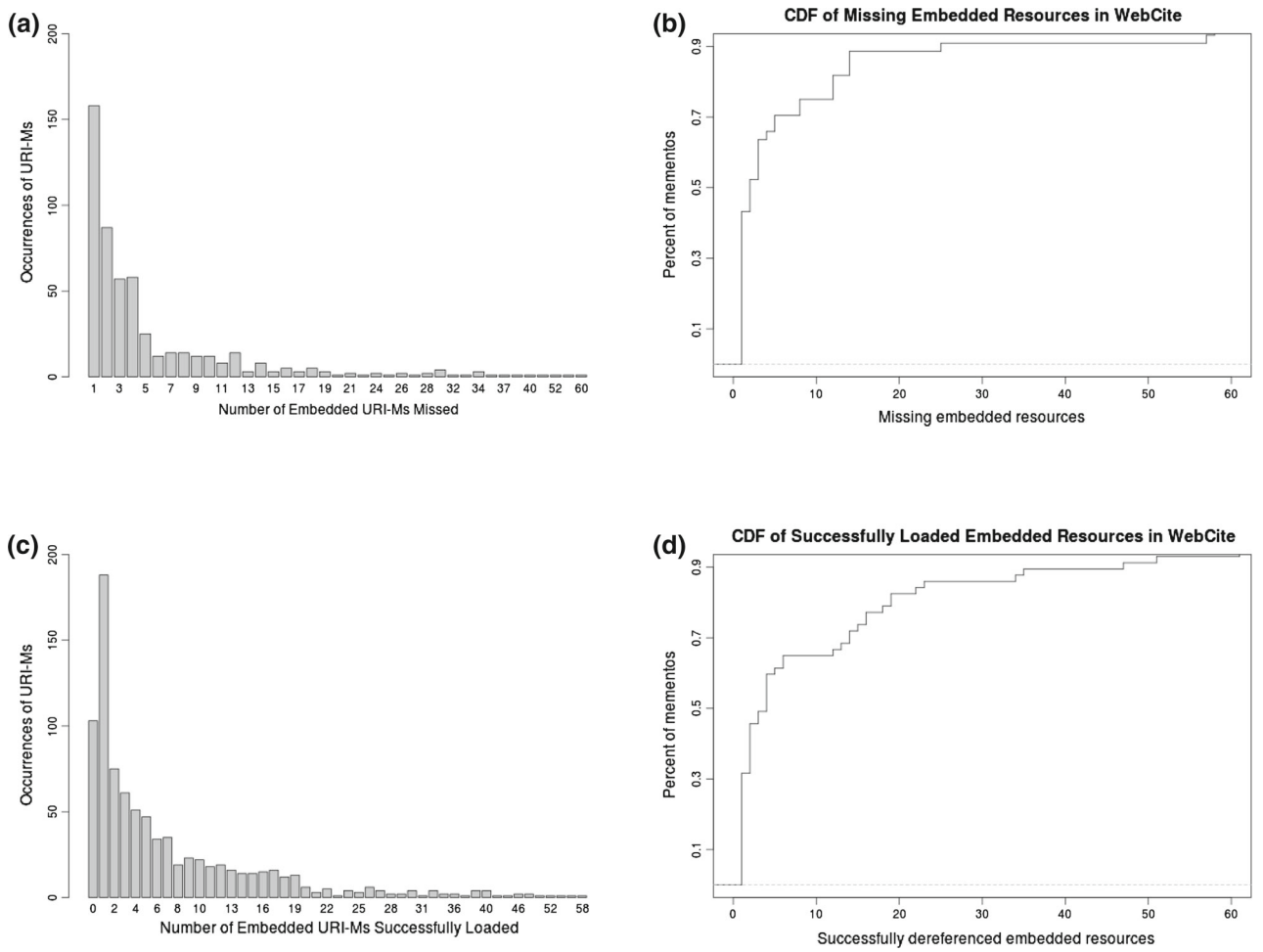

Fig. 9 The distribution of the number of successfully dereferenced and missing embedded resources per URI-M in WebCite. Note that we limited the figures to 60 missing or successfully dereferenced embedded resources, respectively. **a** Occurrences of missing embedded resource numbers in WebCite as a histogram. **b** Distribution of missing embedded resources within the collection of WebCite mementos as a CDF. **c** Occurrences of successfully dereferenced embedded resource numbers in WebCite as a histogram. **d** Distribution of successfully dereferenced embedded resources within the collection of WebCite mementos as a CDF

redirected. We then classified the 50 URI-Rs as having deferred representations and crawled the set of URIs with Heritrix and PhantomJS.

During the Heritrix crawl, we used the 50 URI-Rs as a set of seed URIs and allowed Heritrix to create their mementos. The final frontier size of this crawl was 1,588 URIs of embedded resources used to create the mementos. Using our damage algorithm, we measured the damage of the mementos created by Heritrix and found that $\overline{D_m} = 0.148$. Recall that for the Internet Archive, $\overline{D_m} = 0.13$.

To ensure the crawler executes JavaScript and captures JavaScript-dependent resources during the creation of mementos, we then crawled the 50 URI-Rs with PhantomJS. We recorded the embedded resources needed to create the representation, including those originating from JavaScript. This created a frontier of 3,364 URIs which we used as a seed URI list in Heritrix. We then used Heritrix to create the mementos using only the seed URI list, effectively creating mementos using the frontier list of PhantomJS. For this crawl, $\overline{D_m} = 0.1291$.

PhantomJS provided a 13.5 % improvement to the collection damage over Heritrix. This provides further evidence that JavaScript-dependent representations reduce the quality of mementos due to traditional crawlers' inability to execute JavaScript.

Not only does using PhantomJS provide a larger crawl frontier, but the damage rating of the resulting mementos is lower. In short, this initial investigation suggests that using PhantomJS mitigates the impact of JavaScript on resources with deferred representations and results in higher-quality mementos.

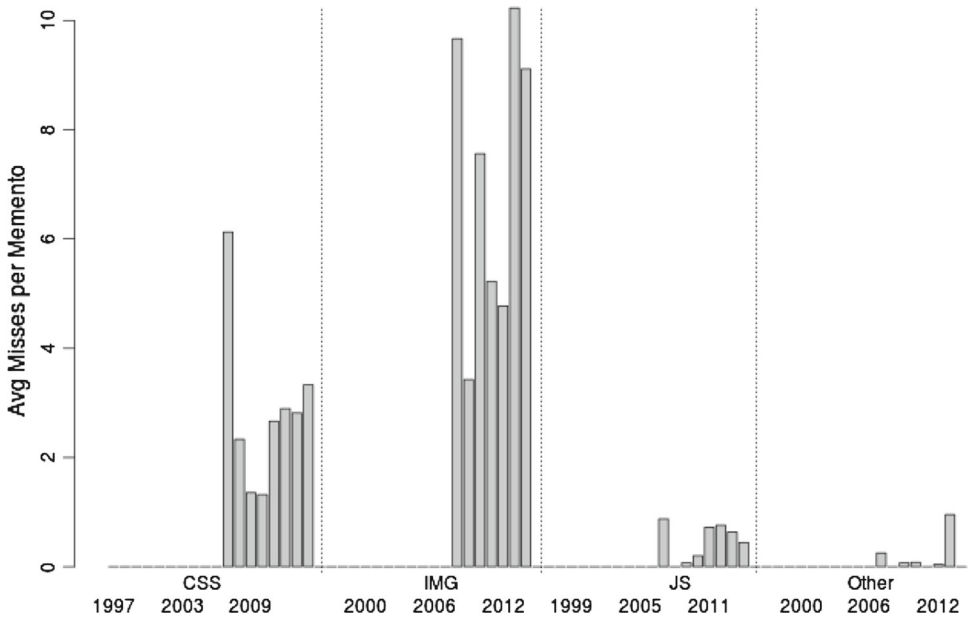

Fig. 10 The number of missed embedded resources per WebCite memento per year and MIME type

9 Measuring Archive.today

Archive.today [3] is another page-at-a-time archival service like WebCite. Archive.today and WebCite were established for different purposes, each offering its own benefits. WebCite was established for the purpose of archiving pages that appear in scholary publications [23], although its use has since expanded to the general Web. Archive.today was established later and with a more modern technology base with respect to JavaScript and Ajax, and always had a focus on the general Web user. Archive.today does not archive resources such as PDFs or XML, while WebCite makes an attempt to archive such resources.

While WebCite does not properly archive deferred representations, Archive.today creates mementos that limit leakage [13,28] (leakage occurs when a memento improperly embeds live Web resources, often through JavaScript) and missing embedded resources typically occurring in other archival services that ignore JavaScript. We leave the decision as to which service is better under what conditions as an exercise for the reader. However, in an effort to study the impact of Archive.today's handling of JavaScript on D_m, we submitted each of our 1,861 URI-Rs to Archive.today for archiving to create mementos of each resource.

When Archive.today creates a memento, it modifies the DOM to remove references to embedded resources that were not available at archive time (i.e., embedded resources that returned a non-200 HTTP response code) [29]. This results in a memento that—if created properly—has no missing embedded resources. Additionally, Archive.today obfuscates the URIs of embedded mementos, preventing a reliable mapping from URI-R to URI-M. For example, the live resource might have an embedded image such as

```
<img src=``http://d3n8a8pro7vhmx.cloudfront.
    net/peoplesgrocery/sites/1/meta_images/
    large/pg_sidebar_logo.png?1372698696''
    width=``315'' height=``315''>
```

and Archive.today will convert the URI-R to the following URI-M:

```
<img src=``https://archive.today/v9cDq/
e632aee8994b72b42e8f7a977ddc1cb63329d9f5''
style=``text-align:left;box-sizing: border-box;
 ... ''/>
```

Due to these two archival practices, the damage algorithm used in this paper is ineffective for determining memento quality. For this reason, we alter the method of measuring the effectiveness of Archive.today's archival process.

We initiated the archiving of each URI-R in our collection by Archive.today. We counted the number of embedded resources that were successfully loaded into the live resources (i.e., returned an HTTP 200 response when their URIs were dereferenced) and compared this number to the number of embedded resources successfully archived by Archive.today, resulting in a delta between live resource and memento embedded resources that we will refer to as Δ_m. It is worth noting that the delta between the number of embedded resources in live resources and mementos (Δ_m) is a measure of neither M_m nor D_m, but is instead a mechanism for understanding memento fidelity.

We found that Archive.today has a $\overline{\Delta_m} = 19.9$ ($\sigma = 39.2$), meaning that on average, Archive.today did not archive 19.9 embedded resources from the live page due to either its inability to archive the resources, or because Archive.today may

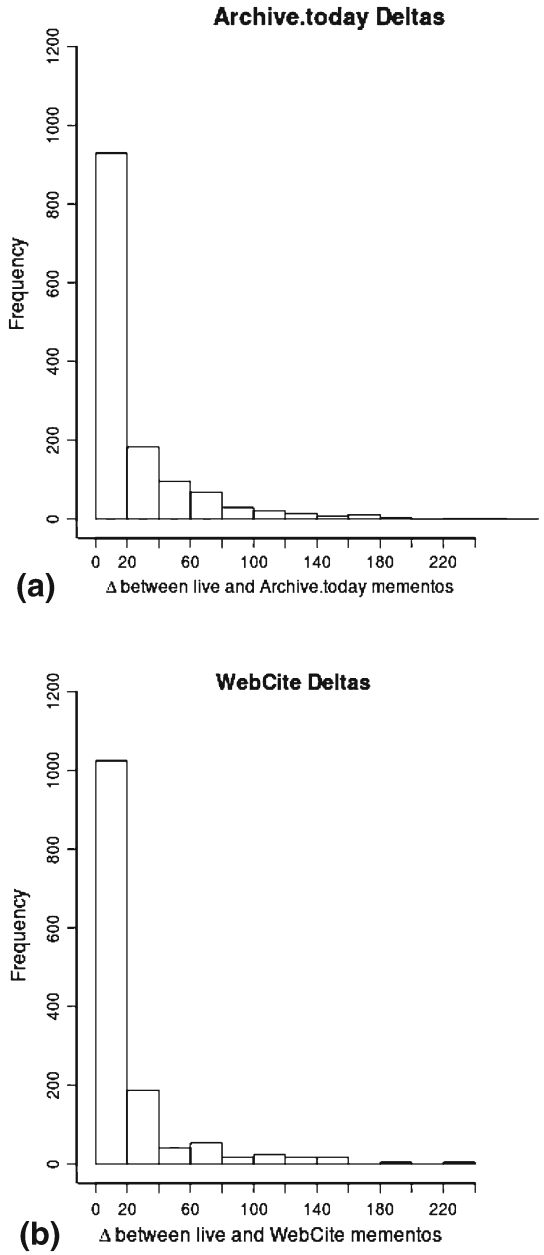

Fig. 11 The Δ_m measurements of Archive.today and WebCite indicate that Archive.today creates higher fidelity mementos than WebCite. **a** Histogram of the memento versus live resource Δ_m in Archive.today. **b** Histogram of the memento versus live resource Δ_m in WebCite

have deemed the embedded resources not suitable for archiving[9]. A histogram of all Δ_m measures is provided in Fig. 11a.

We submitted each URI-R in the collection to WebCite and recorded Δ_m for the WebCite mementos in the exact way we measured Δ_m for Archive.today. In this way, we can compare the two page-at-a-time archivers to determine which service creates higher fidelity mementos. WebCite has a $\overline{\Delta_m} = 21.6$ ($\sigma = 41.7$), which is higher than the $\overline{\Delta_m}$ of Archive.today. The histogram of the WebCite Δ_m is provided in Fig. 11b. The higher WebCite $\overline{\Delta_m}$ indicates that Archive.today creates higher fidelity mementos than WebCite, likely due to its superior support of JavaScript-dependent representations.

10 Conclusions

In this paper, we demonstrated that Web users (as represented by Mechanical Turk workers) can correctly identify undamaged mementos[10] (m_0 vs m_1) 81 % of the time when presented with an original and a manually damaged pair of mementos. After randomly selecting 100 URI-Ms from the Internet Archive TimeMaps of 1,861 URI-Rs, we show that turkers' assessment of damage does not match that of M_m; in fact, their perception of damage more closely aligns with a random selection than with M_m.

To provide a damage metric closer to the perception of Web users, we proposed D_m, a damage calculation algorithm that estimates embedded resource importance to determine the perceived damage of mementos. Using turker evaluations, we showed that D_m aligns with turker perception 32 % of the time when considering all ΔD_m values—an improvement of 17 % over M_m. If we limit $\Delta D_m > 0.30$, we achieve an agreement of 71 %, an improvement of 51 % over M_m. We show that the performance of D_m is closer to that of the m_0 versus m_1 test than both M_m and a random selection.

We used D_m to measure the performance of the Internet Archive by measuring $\overline{D_m}$ of 1,861 URI-Rs. The average damage of the Internet Archive collection is 0.13 per memento and is missing 15 % of its embedded resources. Mementos are missing 2.05 important resources on average. The Internet Archive has gotten better at mitigating damage over time, reducing D_m from 0.16 (1998) to 0.13 (2013).

Page-at-a-time archivers perform differently than the Internet Archive. We measured mementos of our collection in WebCite, finding that the average damage of the collection is 0.397 per memento and is missing 18 % of its embedded resources. Mementos are missing 10.1 resources on average. Even though damage in the Internet Archive is improving, the damage in WebCite is getting worse, increasing D_m from 0.375 (2007) to 0.475 (2013).

We also demonstrate that JavaScript-dependent representations have a detrimental impact on D_m and M_m. By using a crawl strategy in which JavaScript is executed during the

[9] Archive.today lists the resources it saves and does not save in its FAQ page at http://archive.today/faq.html.

[10] "Undamaged" mementos are mementos without purposefully removed embedded resources. Note that some live Web resources may have damage because they are missing embedded resources, and this damage is reflected in our undamaged and subsequently intentionally damaged mementos.

crawl, damage in the resulting mementos can be improved by 13.5 %.

With D_m, archival services can evaluate their performance and the quality of their mementos. The archives could measure a selection of mementos (either randomly sampled or by identifying those missing a proportion of embedded resources, such as $\Delta D_m > 0.30$) for damage to determine whether or not they have been satisfactorily archived. That is, with this algorithm, the archives can provide the greatest damage improvement through targeted repair efforts (e.g., identify mementos that require additional attention to ensure proper archiving). Archives can also use historical damage ratings of a URI-R to identify memento improvements or changes.

We also measured the damage of mementos in WebCite and demonstrated that the damage in the Internet Archive ($\overline{D_m} = 0.128$) is less than that in WebCite ($\overline{D_m} = 0.397$). We know from previous works that WebCite does not archive JavaScript-dependent representations easily. We also measured Archive.today to determine the fidelity of an archival service that makes an effort to use headless browsing to capture JavaScript-dependent representations. We found that Archive.today had a delta of 19.9 embedded resources between mementos and live resources, while WebCite had a delta of 21.6. This shows that Archive.today provides a higher fidelity memento than WebCite.

This is a preliminary investigation of memento damage. We have shown that percentage of embedded resources missing is not an accurate representation of damage and have proposed a more accurate metric. Our future work will continue to improve upon the metric by using larger datasets, more turkers, and machine learning to further hone D_m. This will include a refinement of the relative weights of the embedded resources (e.g., the relative importance of CSS vs images). We will also investigate the cumulative damage rating over time. For example, a logo that never changes over a 5-year period could have increased importance due to its use over multiple mementos. We plan to also measure the damage improvement of mementos if embedded resources are retroactively captured and included in past mementos. This cumulative damage improvement can help identify embedded resources that should be targeted by archives.

Acknowledgments This work was supported in part by the National Science Foundation (NSF) (IIS 1009392), the Library of Congress, and the National Endowment for the Humanities (NEH) Digital Humanities Implementation Grant (DHIG) (HK-50181-14).

References

1. Ainsworth, S.G., Nelson, M.L.: Evaluating sliding and sticky target policies by measuring temporal drift in acyclic walks through a web archive. Int. J. Digit. Librar. 1–16 (2014). doi:10.1007/s00799-014-0120-4
2. Alnoamany, Y., Alsum, A., Weigle, M., Nelson, M.: Who and what links to the internet archive. In: Proceedings of the Third International Conference on Theory and Practice of Digital Libraries, pp. 346–357. ACM (2013). doi:10.1007/978-3-642-40501-3_35
3. Archive.today: Archive.today (2013). http://archive.today/
4. Ayala, B.R., Phillips, M.E., Ko, L.: Technical report. Current Quality Assurance Practices in Web Archiving (2014)
5. Banos, V., Manolopoulos, Y.: A Quantitative approach to evaluate website archivability using the CLEAR+ Method. Int. J. Digit. Librar. 1–24 (2015). http://link.springer.com/article/10.1007%2Fs00799-015-0144-4
6. Banos, V., Yunhyong, K., Ross, S., Manolopoulos, Y.: CLEAR: A credible method to evaluate website archivability. In: Proceedings of the 9th International Conference on Preservation of Digital Objects (2013)
7. Ben Saad, M., Ganarski, S.: Archiving the web using page changes patterns: A case study. In: Proceedings of the 11th Annual International Joint Conference on Digital Libraries, pp. 113–122 (2011). doi:10.1145/1998076.1998098
8. Ben Saad, M., Ganarski, S.: Archiving the web using page changes patterns: a case study. Int. J. Digit. Libr. **13**(1), 33–49 (2012). doi:10.1007/s00799-012-0094-z
9. Ben Saad, M., Pehlivan, Z., Ganarski, S.: Coherence-oriented crawling and navigation using patterns for web archives. In: Proceedings of the First International Conference on Theory and Practice of Digital Libraries, pp. 421–433 (2011)
10. Brunelle, J.F.: Google and JavaScript. http://ws-dl.blogspot.com/2014/06/2014-06-18-google-and-javascript.html (2014)
11. Brunelle, J.F.: Fixing links on the live web, breaking them in the archive. http://ws-dl.blogspot.com/2015/02/2015-02-17-fixing-links-on-live-web.html (2015)
12. Brunelle, J.F., Kelly, M., Weigle, M.C., Nelson, M.L.: The Impact of JavaScript on archivability. Int. J. Digit. Libr. 1–23 (2015). doi:10.1007/s00799-015-0140-8
13. Brunelle, J.F., Nelson, M.L.: Zombies in the archives. http://ws-dl.blogspot.com/2012/10/2012-10-10-zombies-in-archives.html (2012)
14. Denev, D., Mazeika, A., Spaniol, M., Weikum, G.: SHARC: framework for quality-conscious web archiving. In: Proceedings of the 35th International Conference on Very Large Data Bases 2, pp. 586–597 (2009). doi:10.1007/s00778-011-0219-9
15. Eysenbach, G., Trudel, M.: Going, going, still there: using the WebCite service to permanently archive cited web pages. J. Med. Internet Res. 7(5) (2005). doi:10.2196/jmir.7.5.e60
16. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**(8), 861–874 (2006). doi:10.1016/j.patrec.2005.10.010
17. Fersini, E., Messina, E., Archetti, F.: Enhancing web page classification through image-block importance analysis. Inf. Process. Manag. **44**(4), 1431–1447 (2008). doi:10.1016/j.ipm.2007.11.003
18. GNU: Introduction to GNU Wget. http://www.gnu.org/software/wget/ (2013)
19. Gray, G., Martin, S.: Choosing a sustainable web archiving method: A comparison of capture quality. D-Lib Mag. 19(5) (2013). doi:10.1045/may2013-gray
20. Howell, B.A.: Proving web history: how to use the internet archive. J. Internet Law **9**(8), 3–9 (2006)
21. Jack, P.: ExtractorHTML Extract-JavaScript. https://webarchive.jira.com/wiki/display/Heritrix/ExtractorHTML+extract-javascript
22. Kelly, M., Brunelle, J.F., Weigle, M.C., Nelson, M.L.: On the change in archivability of websites over time. In: Proceedings of the Third International Conference on Theory and Practice of Digital Libraries, pp. 35–47 (2013). doi:10.1007/978-3-642-40501-3_5
23. Klein, M., Van de Sompel, H., Sanderson, R., Shankar, H., Balakireva, L., Zhou, K., Tobin, R.: Scholarly context not found: one

in five articles suffers from reference rot. PLoS One **9**(12), e115,253 (2014). doi:10.1371/journal.pone.0115253
24. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 441–450 (2010). doi:10.1145/1718487.1718542
25. Marshall, C.C., Shipman, F.M.: On the institutional archiving of social media. In: Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 1–10 (2012). doi:10.1145/2232817.2232819
26. Mohr, G., Kimpton, M., Stack, M., Ranitovic, I.: Introduction to Heritrix, an archival quality web crawler. In: Proceedings of the 4th International Web Archiving Workshop (2004)
27. Negulescu, K.C.: Web archiving @ the internet archive. Presentation at the 2010 Digital Preservation Partners Meeting, 2010 http://www.digitalpreservation.gov/meetings/documents/ndiipp10/NDIIPP072110FinalIA.ppt
28. Nelson, M.L.: Archive.is supports memento. http://ws-dl.blogspot.com/2013/07/2013-07-09-archiveis-supports-memento.html (2013)
29. Nelson, M.L.: 2014–07-14: "Refresh" For Zombies, Time Jumps. http://ws-dl.blogspot.com/2014/07/2014-07-14-refresh-for-zombies-time.html (2014)
30. PhantomJS: PhantomJS. http://phantomjs.org/ (2013)
31. Rademacher, P., Lengyel, J., Cutrell, E., Whitted, T.: Measuring the perception of visual realism in images. In: Rendering Techniques 2001, Eurographics, p. 235–247. Springer (2001). doi:10.1007/978-3-7091-6242-2_22
32. Reed, S.: Introduction to umbra. https://webarchive.jira.com/wiki/display/ARIH/Introduction+to+Umbra (2014)
33. Rossi, A.: Fixing broken links on the internet. https://blog.archive.org/2013/10/25/fixing-broken-links/ (2013)
34. SalahEldeen, H.M., Nelson, M.L.: Losing my revolution: how many resources shared on social media have been lost? In: Proceedings of the Second International Conference on Theory and Practice of Digital Libraries, pp. 125–137 (2012). doi:10.1007/978-3-642-33290-6_14
35. SalahEldeen, H.M., Nelson, M.L.: Reading the correct history?: Modeling temporal intention in resource sharing. In: Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, pp. 257–266 (2013)
36. SalahEldeen, H.M., Nelson, M.L.: Resurrecting my revolution: Using social link neighborhood in bringing context to the disappearing web. In: Proceedings of the Third International Conference on Theory and Practice of Digital Libraries, pp. 333–345 (2013). doi:10.1007/978-3-642-40501-3_34
37. Sigursson, K.: Incremental crawling with Heritrix. In: Proceedings of the 5th International Web Archiving Workshop (2005)
38. Singh, R., Bhhatarai, B.D.: Information-theoretic identification of content pages for analyzing user information needs and actions on the multimedia web. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 1806–1810 (2009). doi:10.1145/1529282.1529686
39. Song, R., Liu, H., Wen, J.R., Ma, W.Y.: Learning block importance models for web pages. In: Proceedings of the 13th International Conference on World Wide Web, pp. 203–211 (2004). doi:10.1145/988672.988700
40. Spaniol, M., Denev, D., Mazeika, A., Weikum, G., Senellart, P.: Data quality in web archiving. In: Proceedings of the 3rd Workshop on Information Credibility on the Web, pp. 19–26. ACM (2009)
41. Spaniol, M., Mazeika, A., Denev, D., Weikum, G.: Catch me if you can: Visual analysis of coherence defects in web archiving. In: Proceedings of The 9th International Web Archiving Workshop, pp. 27–37 (2009)
42. Sun, Y., Zhuang, Z., Giles, C.L.: A large-scale study of robots.txt. In: Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp. 1123–1124 (2007)
43. Tofel, B.: 'Wayback' for accessing web archives. In: Proceedings of the 7th International Web Archiving Workshop (2007)
44. Van de Sompel, H., Nelson, M.L., Sanderson, R., Balakireva, L.L., Ainsworth, S., Shankar, H.: Memento: Time travel for the websites technical report. arXiv:0911.1112, Los Alamos National Laboratory (2009)
45. Yi, L., Liu, B., Li, X.: Eliminating noisy information in web pages for data mining. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 296–305 (2003). doi:10.1145/956750.956785
46. Zhang, X., Lin, W., Xue, P.: Just-noticeable difference estimation with pixels in images. J. Vis. Commun. Image Represent. **19**(1), 30–41 (2008). doi:10.1109/TMM.2013.2268053