International Journal on Digital Libraries (2022) 23:5–17
https://doi.org/10.1007/s00799-021-00315-w

An extended analysis of the persistence of persistent identifiers of the scholarly web

Martin Klein[1] · **Lyudmila Balakireva**[1]

Received: 8 February 2021 / Revised: 23 September 2021 / Accepted: 1 October 2021 / Published online: 22 October 2021

Abstract
Scholarly resources, just like any other resources on the web, are subject to reference rot as they frequently disappear or significantly change over time. Digital Object Identifiers (DOIs) are commonplace to persistently identify scholarly resources and have become the de facto standard for citing them. This paper is an extended version of work previously published in the proceedings of the 2020 International Conference on Theory and Practice of Digital Libraries (TPDL). We investigate the notion of persistence of DOIs by conducting a series of experiments to analyze a DOI's resolution on the web, with this work presenting a set of novel investigations to expand on our previous work. We derive confidence in the persistence of these identifiers in part from the assumption that dereferencing a DOI will consistently return the same response, regardless of which HTTP request method we use or from which network environment we send the requests. Our experiments show, however, that persistence, according to our interpretation, is not warranted. We find that scholarly content providers respond differently to varying request methods and network environments, change their response to requests against the same DOI, and even return inconsistent results over a period of time. We present the results of our quantitative analysis that is aimed at informing the scholarly communication community about this disconcerting lack of consistency.

Keywords Digital Object Identifiers (DOIs) · HTTP resolution · Scholarly communication

1 Introduction

The web is a very dynamic medium where resources frequently are being created, deleted, and moved [2,7,8]. Scholars have realized that, due to this dynamic nature, reliably linking and citing scholarly web resources are not a trivial matter [18,19]. Persistent identifiers such as the Digital Object Identifier (DOI)[1] have been introduced to address this issue and have become the de facto standard to persistently identify scholarly resources on the web. The concept behind a DOI is that while the location of a resource on the web may change over time, its identifying DOI remains unchanged and, when dereferenced on the web, continues to resolve to the resource's current location. This concept is based on the underlying assumption that the resource's publisher updates the mapping between the DOI and the resource's location if and when the location has changed. If this mapping is reliably maintained, DOIs indeed provide a more persistent way of linking and citing web resources.

While this system is not perfect [3] and we have previously shown that authors of scholarly articles often do not utilize DOIs where they should [23], DOIs have become an integral part of the scholarly communication landscape.[2] Our work is motivated by questions related to the consistency of resolving DOIs to scholarly content. From past experience crawling the scholarly web, for example, in [11,17], we have noticed that publishers do not necessarily respond consistently to simple HTTP requests against DOIs. We have instead observed scenarios where their response changes depending on what HTTP client and method are used. If we can demonstrate at scale that this behavior is commonplace in the scholarly communication landscape, it would raise significant concerns about the persistence of such identifiers for the scholarly web. In other words, we are driven by the question that if

[1] https://www.doi.org/.

✉ Martin Klein
mklein@lanl.gov

Lyudmila Balakireva
ludab@lanl.gov

[1] Los Alamos National Laboratory, Los Alamos, NM 87545, USA

[2] https://data.crossref.org/reports/statusReport.html.

we cannot trust that requests against the same DOI return the same result, how can we trust in the identifier's persistence?

In our initial study [14], we reported the outcome of our initial investigation into the notion of persistence of DOIs from the perspective of their behavior on the web. We found early indicators for scholarly publishers responding differently to different kinds of HTTP requests against the same DOI. In our more recent work [13], we expand on that study by:

- re-executing the previous experiments with an improved technical setup,
- adding additional experiments from a different network environment,
- adding additional experiments with different access levels to scholarly content, and
- adding a comparison corpus to help interpret our findings and put them into perspective.

In this paper, we further report on additional experimentation that:

- adds a temporal analysis to previous findings based on re-executing experiments at a later time and
- offers a preliminary content similarity analysis of the DOI-identified web resources.

By adding these dimensions to our previous work, we identify two novel research questions ($RQ6$ and $RQ7$) to the previously addressed set:

1. What differences in dereferencing DOIs can we detect and highlight?
2. In what way (if at all) do scholarly content providers' responses change depending on network environments?
3. How do observed inconsistencies compare to responses by web servers providing popular (non-scholarly) web content?
4. What effect do Open Access and non-Open Access content providers have on the overall picture?
5. What is the effect of subscription levels to the observed inconsistencies?
6. When considering a temporal dimension, what differences in dereferencing DOIs can we detect and highlight?
7. To what extent do observed consistencies on the HTTP network level translate to content similarities?

These seven research questions (RQs) aim at a quantitative analysis of the consistency of HTTP responses, consistencies of responses over time, and content similarity of identified web resources. We do not claim that such consistencies are the only factors that contribute to persistence of scholarly resource identifiers. We argue, however, that without a reassuring level of consistency, our trust in the persistence of an identifier and its resolution to a resource's current location is significantly diminished.

In the remainder of this paper, we will highlight previous related work in Sect. 2, outline the experiments' setup in Sect. 3, address our research questions in Sect. 4, discuss some aspects for future work in Sect. 5, and draw our conclusions in Sect. 6.

2 Related work

DOIs are the de facto standard for identifying scholarly resources on the web, supported by traditional scholarly publishers as well as repository platforms such as Figshare and Zenodo, for example. When crawling the scholarly web for the purpose of aggregation, analysis, or archiving, DOIs are therefore often the starting point to access resources of interest. The use of DOIs for references in scholarly articles, however, is not as widespread as it should be. In previous work [23], we have presented evidence that authors often use the URL of a resource's landing page rather than its DOI when citing the resource. This situation is undesirable as it requires unnecessary deduplication for efforts such as metrics analysis or crawling. These findings were confirmed in a large-scale study by Thompson and Jian [22] based on two samples of the web taken from Common Crawl[3] datasets. The authors were motivated to quantify the use of HTTP DOIs versus URLs of landing pages in these two samples generated from two snapshots in time. They found more than 5 million actionable HTTP DOIs in the first dataset from 2014 and about 10% of them in the second dataset from 2017 but identified as the corresponding landing page URL, not the DOI. It is worth noting that not all resources referenced in scholarly articles have a DOI assigned to them and are therefore subject to typical link rot scenarios on the web. In large-scale studies, we have previously investigated and quantified the "reference rot" phenomenon in scholarly communication [11,17] focusing on "web at large" resources that do not have an identifying DOI.

Any large-scale analysis of the persistence of scholarly resources requires machine access as human evaluations typically do not scale. Hence, making web servers that serve (scholarly) content more friendly to machines has been the focus of previous efforts by the digital library community with the agreement that providing accurate and machine-readable metadata is a core requirement [5,20]. To support these efforts, recently standardized frameworks are designed to help machines synchronize metadata and content between scholarly platforms and repositories [16].

[3] http://commoncrawl.org/.

The study by Alam et al. [1] is related to ours in the way that the authors investigate the support of various HTTP request methods by web servers serving popular web pages. The authors issue OPTIONS requests and analyze the values of the "Allow" response header to evaluate which HTTP methods are supported by a web server. The authors conclude that a sizable number of web servers inaccurately report supported HTTP request methods.

As mentioned, this paper builds upon our previous work [14] and represents an extension to the TPDL 2020 conference proceedings [13]. Novel aspects presented here are related to a temporal aspect of dereferencing DOIs and a preliminary content analysis of the representations of the redirect chains' final link. Temporal aspects of the web have been studied in the past, for example by Bordiono et al. [4] on the UK Web, by Radinsky et al. [21] in the context of its implications to information retrieval tasks, and Buriol et al. [6] to assess temporal changes in specific domains, in this case changes to the Wikipedia graph. Our chosen methods for content similarity analysis are well established and have been successfully applied in the past [15].

3 Experimental setup

3.1 Dataset generation

To the best of our knowledge, no dataset of DOIs that identify content representative of the diverse scholarly web is available to researchers. Part of the problem is the scale and diversity of the publishing industry landscape but also the fact that the Science, Technology, and Medicine (STM) market is dominated by a few large publishers [10]. We therefore reuse the dataset generated for our previous work [14] that consists of 10, 000 randomly sampled DOIs [12] from a set of more than 93 million DOIs crawled by the Internet Archive. We refer to [14] for a detailed description of the data gathering process, an analysis of the composition of the dataset, and a discussion of why we consider this dataset to be representative of the scholarly landscape. In addition, to be able to put our findings from the DOI-based dataset in perspective, we created a dataset of the top 10, 000 most popular URIs on the web as extracted from the freely available "Majestic Million" index[4] on November 14, 2019.

3.2 HTTP requests, clients, and environments

HTTP transactions on the web consist of a client request and a server response. As detailed in RFC 7231 [9], requests contain a request method and request headers and responses contain corresponding response headers. GET and HEAD are two of the most common HTTP request methods (also detailed in RFC 7231). The main difference between the two methods is that upon receiving a client request with the HEAD method, a server only responds with its response headers but does not return a content body to the client. Upon receiving a client request with the GET method, on the other hand, a server responds by sending the representation of the resource in the response body in addition to the response headers.

It is important to note that, according to RFC 7231, we should expect a server to send the same headers in response to requests against the same resource, regardless whether the request is of type HEAD or GET. RFC 7231 states: "The server SHOULD send the same header fields in response to a HEAD request as it would have sent if the request had been a GET...".

To address our research questions outlined earlier, we utilize the same four methods described in [14] to send HTTP requests:

- **HEAD**, a HEAD request with cURL[5]
- **GET**, a simple GET request with cURL
- **GET+** a GET request that includes typical browsing parameters such as user agent and accepted cookies with cURL
- **Chrome**, a GET request with Chrome[6]

We sent these four requests against the HTTPS-actionable format of a DOI, meaning the form of https://doi.org/<DOI>. This is an important difference to our previous work ([14]) where we did not adhere to the format recommended by the DOI Handbook.[7] For the first set of experiments and to address RQ1, we send these four HTTP requests against each of the 10, 000 DOIs from an Amazon Web Services (AWS) virtual machine located at the US East Coast. The clients sending the requests are therefore not affiliated with our home institution's network. Going forward, we refer to this external setup as the DOI_{ext} corpus. In addressing RQ2, we anticipate possible discrepancies in HTTP responses from servers depending on the network from which the request is sent. Hence, for the second set of experiments, we send the same four requests to the same 10, 000 DOIs from a machine hosted within our institution's network. Given that the machine's IP address falls into a range that conveys certain institutional subscription and licensing levels to scholarly publishers, this internal setup, which we refer to going forward as DOI_{int}, should help surface possible

[4] https://blog.majestic.com/development/majestic-million-csv-daily/.

[5] A popular lightweight HTTP client for the command line interface https://curl.haxx.se/.

[6] Web browser controlled via the Selenium WebDriver https://selenium.dev/projects/.

[7] https://www.doi.org/doi_handbook/3_Resolution.html.

differences. To address RQ3, we compare our findings to responses from servers providing non-scholarly content by sending the same four requests against each of the 10,000 URIs from our dataset of popular websites. From here on, we refer to this corpus as the *Web* dataset.

4 Experimental results

In this section, we report our observations when dereferencing HTTPS-actionable DOIs with our four methods. Each method automatically follows HTTP redirects and records information about each link in the redirect chain. For example, a HEAD request against https://doi.org/10.1007/978-3-030-30760-8_15 results in a redirect chain consisting of the following links:

1. http://link.springer.com/10.1007/978-3-030-30760-8_15
2. https://link.springer.com/10.1007/978-3-030-30760-8_15
3. https://link.springer.com/chapter/10.1007%2F978-3-030-30760-8

with the last one showing the 200 *OK* response code. Note that only the first redirect comes from the server at doi.org (operated by the Corporation for National Research Initiatives (CNRI)[8]) and it points to the appropriate location on the publisher's end. All consecutive redirects remain in the same domain and, unlike the HTTP DOI, are controlled by the publisher.

It is important to note that all four methods are sent with the default timeout of 30 seconds, meaning the request times out if a server does not respond within this time frame. In addition, all methods are configured to follow a maximum of 20 redirects.

4.1 Final response codes

The first aspect of consistency, as projected onto our notion of persistence, we investigate is the response code of the last accessible link in the redirect chain when dereferencing DOIs (or URIs in the case of the *Web* corpus). Intuitively and informed by our understanding of persistence, we expect DOIs as persistent identifiers return the same response code to all issued requests, regardless of the request method used.

Table 1 summarizes the response codes for our three different corpora and the four different methods for each of them. The frequency of response codes (in percent) is clustered into 200-, 300-, 400-, and 500-level columns, plus an error column. The latter represents requests that timed out and did not return any response or response code. The first main observation from Table 1 is that the ratio of response codes for all four methods and across all three corpora is inconsistent. Even within individual corpora, we notice significant differences. For example, for the DOI_{ext} corpus we see 40% and 24% of GET and GET+ requests, respectively, end in 300-level response codes. We consider this number particularly high as the vast majority of these responses have a 302 *Found* status code that indicates further action needs to be taken by the client to fulfill the request, for example, send a follow-up request against the URI provided in the Location header field (see RFC 7231 [9]). In other words, no HTTP request (and redirect chain) should end with such a response code. A different reason for these observations could be a server responding with too many consecutive 300-level responses, causing the client to stop making follow-up requests. (The default for our methods was 20 requests.) However, we only recorded this behavior a few times and it therefore cannot explain these high numbers. Another observation for the same corpus is the fairly high ratios for 400-level responses, particularly for HEAD requests. The fact that this number (12.58%) is two to three times as high as for the other three requests for the same corpus is noteworthy.

Table 1 Percentage of final HTTP response codes, aggregated into five levels, following the DOI/URI redirect chain

Corpus	Method	2xx	3xx	4xx	5xx	Err
DOI_{ext}	HEAD	75.4	9.93	12.58	2.09	0
	GET	53.07	40.49	6.06	0.06	0.32
	GET+	70.71	24.34	4.58	0.05	0.32
	Chrome	87.79	6.17	5.94	0.1	0
DOI_{int}	HEAD	70.64	16.98	8.85	3.52	0.01
	GET	76.13	16.66	5.71	1.48	0.02
	GET+	80.29	15.26	4.04	0.41	0
	Chrome	90.2	5.95	3.57	0.18	0.1
Web	HEAD	70.69	4.86	5.63	1.32	17.5
	GET	56.71	5.35	2.78	0.6	34.56
	GET+	57.43	5.54	1.87	0.52	34.64
	Chrome	74.8	4.56	2.66	0.65	17.33

Except for HEAD requests, the ratio of 300-level responses decreased for the DOI_{int} corpus. We do see more 301 *Moved Permanently* responses in this corpus compared to DOI_{ext}, but given that this fact should not have a different impact for individual request methods, we can only speculate why the ratio for HEAD requests went up. The ratio of 400-level responses is not insignificant in both corpora, and it is worth noting that this category is dominated by the 403 response code, which means a server indicates to a client that access to the requested URI is forbidden. This response would make sense for requests to resources for which we do not have

[8] https://www.cnri.reston.va.us/.

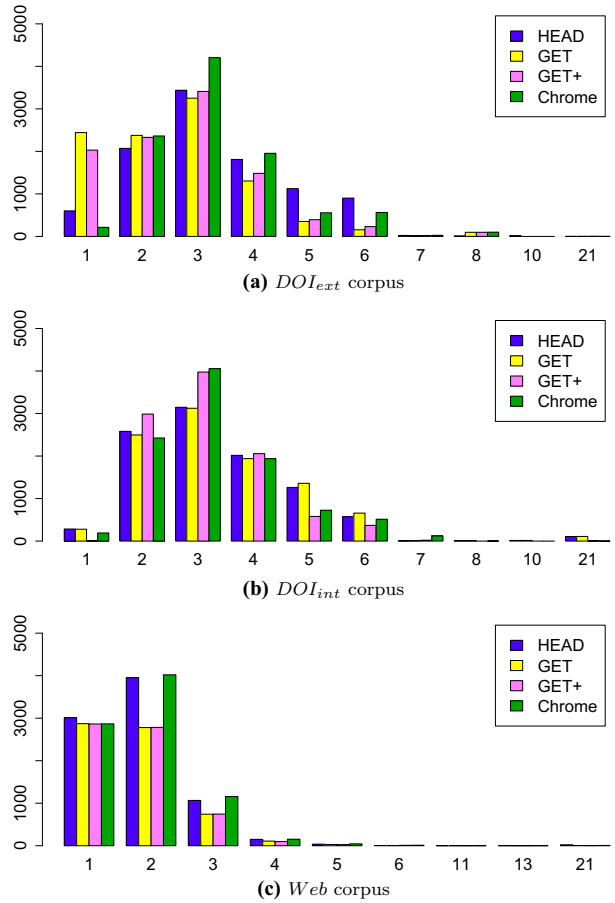


Fig. 1 Frequency (y-axes) of number of total links in DOI/URI redirect chains (x-axes) per corpus

institutional subscription rights or licensing agreements, for example, but then we would expect to see these numbers being consistent for all methods.

As a comparison, the requests for the Web corpus seem to mostly result in one of two columns. Either they return a 200-level response or an error (no response code at all). The ratios in the error category are particularly high for the GET and the GET+ methods at around 34%.

4.2 Redirect chain

The next aspect of persistence in our investigation is the overall length of the redirect chain when dereferencing DOIs. Intuitively speaking, we expect the chain length to be the same for persistent identifiers, regardless of the HTTP method used. Figure 1 shows histograms of chain lengths distinguished by corpora and request methods. Note that the reported lengths are independent of the final response code reported earlier and that DOIs/URIs that resulted in errors are excluded from this analysis. Figure 1a shows the observed chain lengths for the DOI_{ext} corpus. We note that the distribution of chain lengths is not equal among request methods. The GET and GET+ methods, for example, are much more strongly represented at length one than either of the other methods. Generally speaking, however, lengths two, three, and four represent the majority for the requests in the DOI_{ext} corpus.

The same holds true for the DOI_{int} corpus (shown in Fig. 1b), but we notice the frequency of length one has almost disappeared. When comparing the two corpora, we observe that the Chrome method shows fairly consistent frequencies of redirect chain length and most often results in length three.

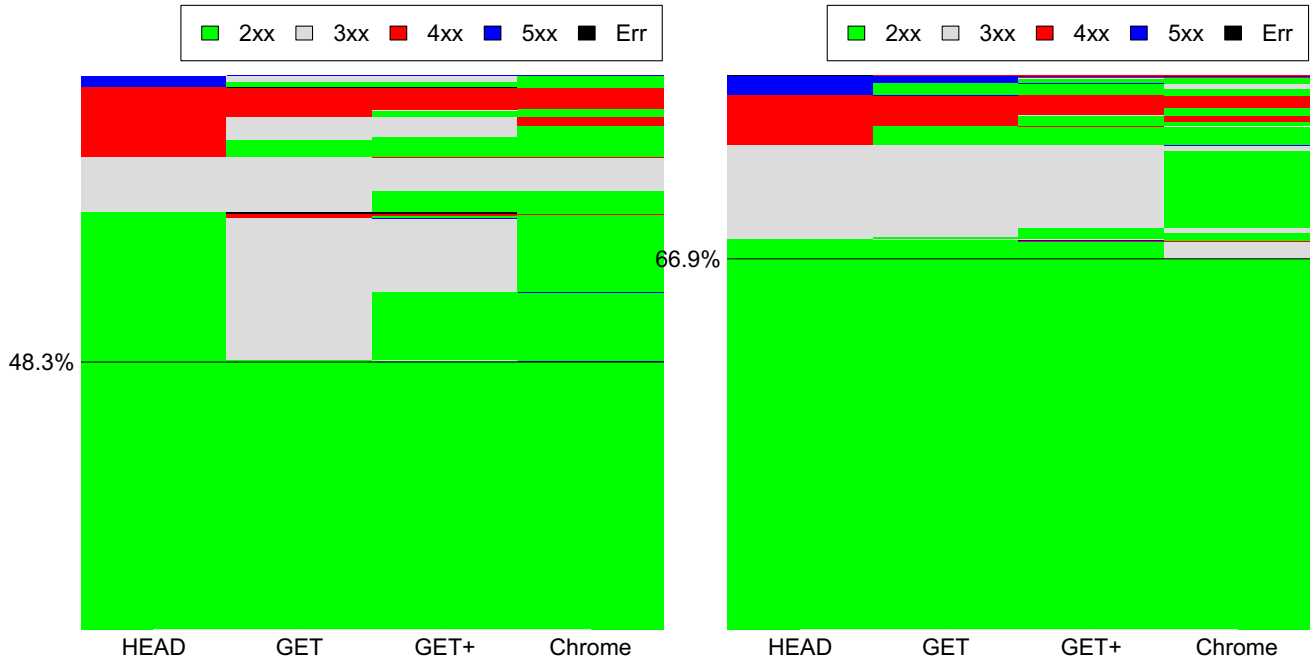
Figure 1c offers a comparison by showing the redirect chain lengths of dereferencing URIs from the Web corpus. We see a significant shift to shorter redirect chains with the majority being of length one or two. While we recorded chains of length four and beyond, these occurrences were much less frequent. The HEAD and Chrome methods appear to be well aligned for all observed lengths.

It is worth mentioning that we recorded chain length beyond our set maximum of 20 (indicated as 21 in the figures). We question the reasoning for such responses but leave a closer analysis of these extensive redirect chains for future work.

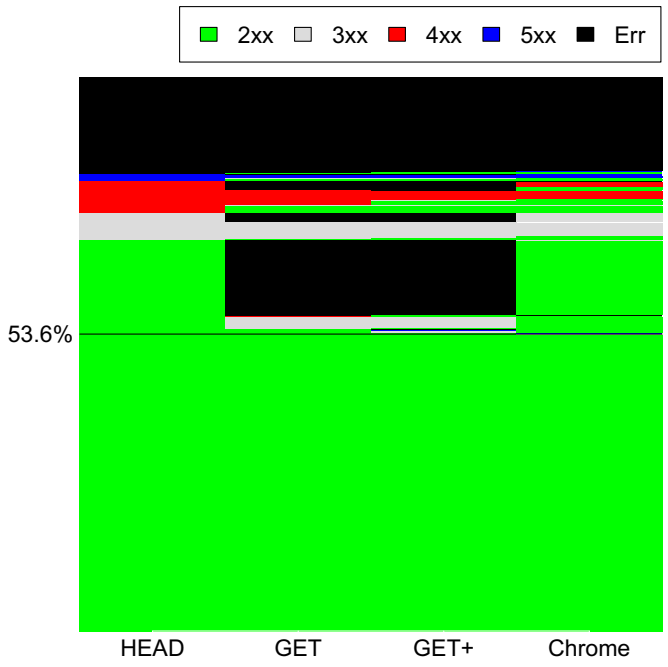
4.3 Changing response codes

The third aspect of our investigation centers around the question whether HTTP response codes change, depending on what HTTP request method is used. We have shown in Sect. 4.1 that dereferencing DOIs does not result in the same response codes but varies depending on what request method we used. In this section, we analyze the nature of response code change per DOI and request method. This investigation aims at providing clarity about if and how response codes change and the ramifications for the notion of persistence.

Figure 2 shows all response codes again binned into 200- (green), 300- (light gray), 400- (red), 500-level (blue), and error (black) responses per DOI for all three corpora. The request methods are represented on the x-axis, and each of the 10, 000 DOIs is displayed on the (unlabeled) y-axis. Figure 2a shows the response codes and their changes from one method to another for the DOI_{ext} corpus. We see that merely 48.3% of all 10, 000 DOIs consistently return a 200-level response, regardless of which request method is used. This number is surprisingly low. The fact that, consistently across request methods, more than half of our DOIs fail to successfully resolve to a target resource strongly indicates that the scholarly communication landscape is lacking the desired level of persistence. We further see major differences in response codes depending on the request method. For example, a large portion, just over 40%, of all DOIs return a 300-level response for the simple GET request. However, 12% of these DOIs return a 200-level response with any of



(a) DOI_{ext} corpus **(b)** DOI_{int} corpus



(c) Web corpus

Fig. 2 Final HTTP response codes by DOI/URI per corpus

the other three request methods and 25% return a 200-level response if only the HEAD or Chrome method is used. We further find 13% of DOIs resulting in a 400-level response with the HEAD request but of these only 30% return the same response for any of the other request methods. In fact, 25% of them return a 200-level response when any other request method is used. Without further analysis of the specific links in the redirect chain and their content, which we leave for future work, we can only hypothesize that web servers of scholarly content take the request method into consideration and respond accordingly when resolving DOIs. However,

this lack of consistency is worrisome for everyone concerned about persistence of the scholarly record.

Figure 2b shows our findings from the DOI_{int} corpus. We see the numbers improved, most noticeably with 66.9% of DOIs returning a 200-level response across the board. However, we still find almost 14% of DOIs returning a 300-level response for the first three and a 200-level response only for our Chrome method. We also see a similar ratio of 400-level responses for the HEAD method that decreases with the GET, GET+, and Chrome methods, similar to our observation for the DOI_{ext} corpus. The ratio of 500-level responses slightly increased from 2% in the previous corpus to 3.5% here. However, here too the majority of those DOIs return a different response code when methods other than HEAD are used. The observations from Fig. 2b show that even requests sent from within a research institution network are treated differently by scholarly content providers and, depending on the request method used, the level of consistency suffers.

Figure 2c shows the numbers for the Web corpus and therefore offers a comparative picture to our above findings. For the Web corpus, we see 53.6% of all 10, 000 URIs returning a 200-level response code, which is ahead of the DOI_{ext} but well below the DOI_{int} corpus numbers. We further see 17% of URIs returning an error, regardless of the request. We can only speculate about the reasons for this high number of unsuccessful requests, but our best guess is that web servers of these popular websites have sophisticated methods in place that detect HTTP requests sent from machines and simply do not send a response when detected. This even holds true for our Chrome method, which closely resembles a human browsing the web. Not unlike what we have seen in the DOI_{ext} corpus the Web corpus shows 15% of requests not being successful with the GET and GET+ methods but being successful (200-level response) with the HEAD and Chrome methods. These findings indicate that popular but not necessarily scholarly content providers also send responses depending on the request method. However, we see fewer 300-, 400-, and 500-level responses for this corpus.

4.4 Responses depending on access level

The distinction between the DOI_{ext} and DOI_{int} corpora serves to highlight patterns for the lack of consistent responses by scholarly publishers when accessed from outside and within an institutional network. Our observations raise further questions about possible differences between access levels. In particular, we are motivated to evaluate the responses for:

- DOIs identifying Open Access (OA) content versus their non-OA counterparts (nOA), addressing RQ4, and
- DOIs identifying content to which we have access due to institutional subscription and licensing agreements (SUB) versus those we do not ($nSUB$), addressing RQ5.

Table 2 Distribution of DOIs leading to OA and nOA resources as well as to SUB and $nSUB$ content in our dataset

	OA	nOA		SUB	nSUB
DOI_{ext}	973	9027	DOI_{int}	1266	8734

We utilize our DOI_{ext} corpus to analyze responses of DOIs identifying OA content and the DOI_{int} corpus to investigate responses for DOIs that lead to licensed content. Identifying OA content can be a non-trivial task, and manual inspection of all of the 10, 000 DOIs is clearly not feasible. We instead utilize the Unpaywall service[9] to determine whether a DOI identifies OA content. The service's API[10] allows for scalable lookup of metadata about scholarly articles identified by a DOI. Part of the obtained metadata records is information about the level of access to an articles, for example, whether it is indeed openly available or not. To identify licensed content, we match institutional subscription information to base URIs of dereferenced DOIs. Table 2 summarizes the resulting numbers of DOIs and their access levels in our corpora. We realize that the numbers for licensed content may not be representative as other institutions likely have different subscription levels to scholarly publishers. However, given that we consider our DOI corpus representative, we are confident the ratios represent a realistic scenario.

Figure 3 shows the final response codes for the DOI_{ext} corpus, similar in style to Fig. 2, with the DOIs along the y-axis and our four request methods on the x-axis. Figure 3a shows the response codes for the 973 OA DOIs, and Fig. 3b shows the remaining 9, 027 DOIs that identify non-OA content. The first observation we can make from these two figures is that OA DOIs return 200-level responses for all requests more often than non-OA DOIs with 59.5% versus 47.1%. We can further see that even for OA DOIs, the GET and GET+ methods do not work well. 26% of DOIs return a 300-level response for these two methods but return a 200-level response for the HEAD and Chrome methods. If we compare Fig. 3 with 2a, we can see a clear resemblance between Fig. 2a, the figure for the overall corpus, and Fig. 3b, the figure for non-OA DOIs. Given the fact that we have many more non-OA DOIs, this may not be all that surprising but it is worth noting that by far the vast majority of 400- and 500-level responses come from non-OA DOIs. Given our dataset, this observation indicates that OA content providers show more consistency across the board compared to non-

[9] https://unpaywall.org/.

[10] https://unpaywall.org/products/api.

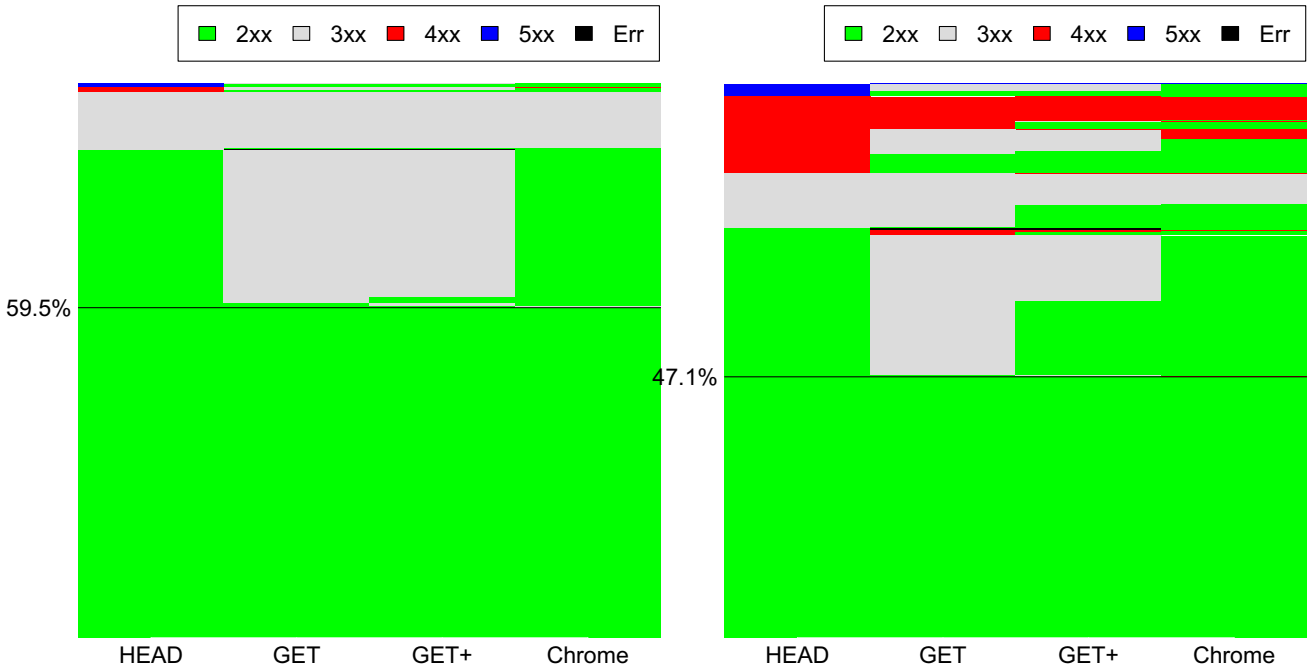


Fig. 3 DOI_{ext} final HTTP response codes distinguished by OA and nOA

OA providers and their positive effect to the overall picture (Fig. 2a) is visible. A larger-scale analysis of OA versus non-OA content providers is needed, however, to more reliably underline this observation. We leave such effort for future work.

Figure 4 shows the final response codes for DOIs that identify institutionally licensed content (Fig. 4a) and content not licensed by our institution (Fig. 4b). We see a much higher ratio of DOIs returning 200-level responses for all request methods for licensed content (84.3%) compared to not licensed content (64.4%). We also notice fewer 300-, 400-, and 500-level responses for licensed content and the Chrome method being almost perfect in returning 200-level responses (99%). When we again compare Fig. 4 to the overall picture for this corpus shown in Fig. 2b, we notice a strong resemblance between Figs. 4b and 2b. This leads us to conclude that providers, when serving licensed content, show more consistency and introduce fewer unsuccessful DOI resolutions.

4.5 Temporal analysis of response codes

In this section, we expand on the experiment outlined in Sect. 4.3 by adding a temporal dimension, addressing RQ6. We repeat dereferencing all 10,000 DOIs with our four request methods from the external network environment nine months after the first run. The comparison of two snapshots of final response codes taken at different points in time (separated by nine months) will provide further insights into the persistence, or lack thereof, when dereferencing DOIs. Our intuitive notion of persistence suggests we should find a very similar, if not identical, picture of the final response codes. While the URI of any of the links in the redirect may change over time, and as such providing good arguments in favor of the DOI principle, the response code of the final link should not change.

Figure 5 provides the overview of final response codes from the DOI_{ext} corpus as observed in November 2020. The figure is structured in the same fashion as previously seen and can therefore directly be compared to Fig. 2a, which shows the results of the same DOI_{ext} corpus as seen in March of 2020. We can make several observations for this temporal analysis. At first glance, Figs. 2a and 5 appear fairly similar, which is what we would expect for running the same experiment setup in the same network environment. However, a closer inspection reveals some noticeable differences. First, the fraction of DOIs that return a 200-level response for all four methods dropped by 3.4%, down from 48.3% to 44.9%. In fact, we notice a drop of 200-level responses for each of the four methods individually. HEAD dropped by 4.5%, GET by 2.6%, GET+ by 4.5%, and even Chrome dropped by 3.5%. On the other hand, we see an increase of 300-level responses for the GET method (up by 11.1% to a total of 45%), 400-level responses for HEAD (up by 23.8% to a total of 16%), and a slight increase of 500-level responses for the Chrome method (total of 1.8%).

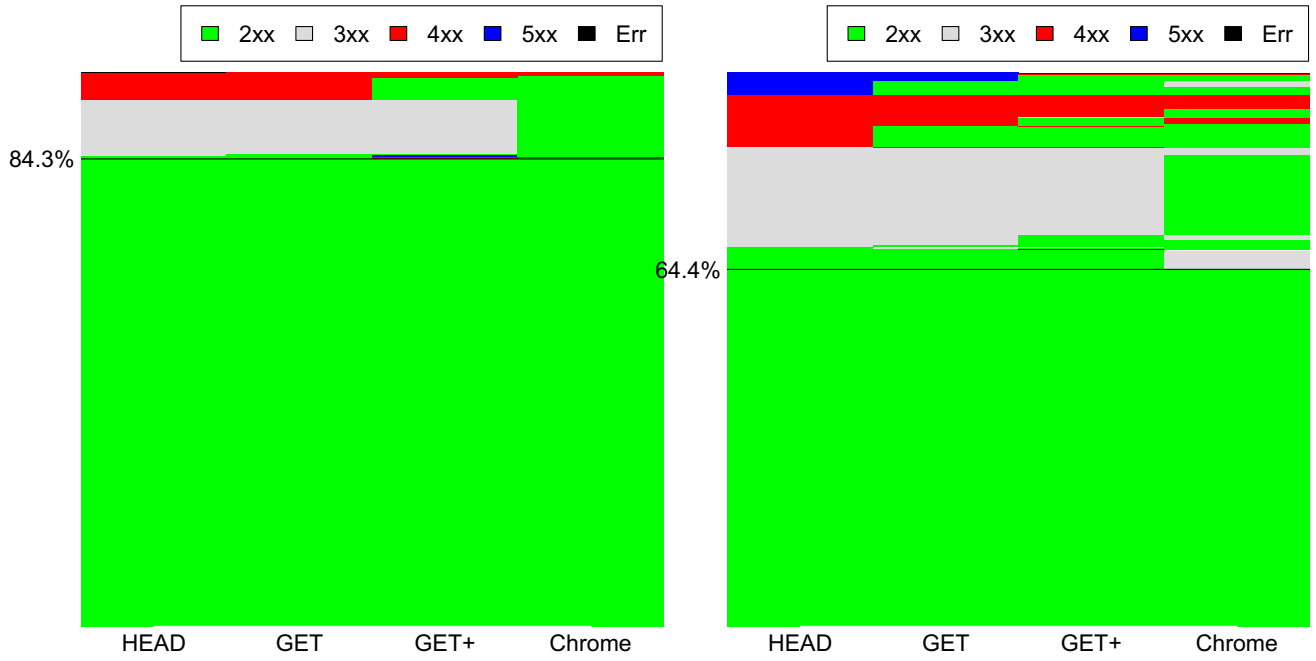


Fig. 4 DOI_{int} final HTTP response codes distinguished by SUB and nSUB

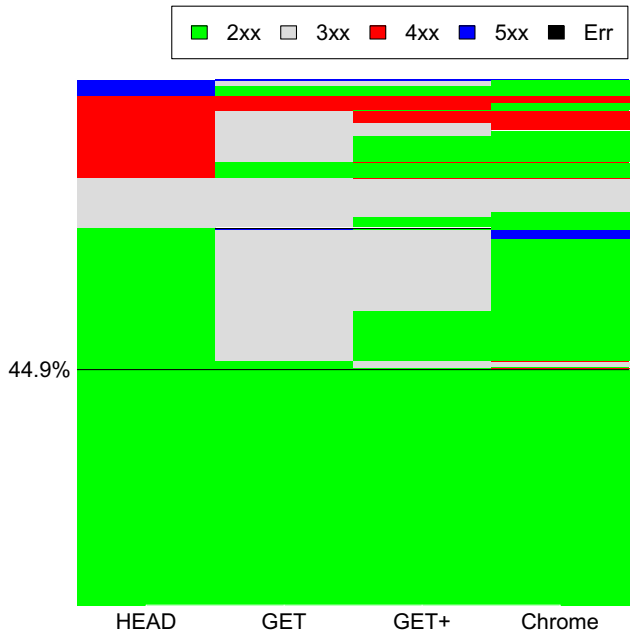


Fig. 5 Final HTTP response codes by DOI for DOI_{ext} corpus in November 2020

More such experiments are needed to confidently draw the conclusion of a pattern of declining 200-level responses, but the numbers do indicate further inconsistencies in dereferencing DOIs, even when taking transient errors and other network failures into account.

4.6 Content similarity of dereferenced resources

Aside from inconsistencies in responding on the HTTP network level, we were further motivated to investigate the content similarity of final links in our DOI redirect chains, addressing RQ7. Even though, given all our previously shown results, we have little confidence in the appropriate use of the HTTP response codes, we cannot reasonably assume that, for example, a 300-level response for one method is meant to convey the same message or content as a 200-level response for another method. We therefore limit our content comparison to those DOIs that return a 200-level response for all four methods consistently. We use the DOI_{ext} corpus generated in November of 2020 and identify the 4, 485 DOIs that meet this criteria within the dataset. Upon analyzing the redirect chains, we noticed a non-insignificant number of DOIs whose redirect chain ended in a different URI for different methods, while returning a 200-level response for all of them. We consider this an intriguing finding in its own right but leave a more comprehensive investigation into this matter for future work. For the purpose of a sensible content comparison, however, we restrict our analysis to those DOIs that consistently return the same URI at the end of their redirect chain for all four methods. Table 3 summarizes the number of DOIs with a matching and non-matching URI for all possible comparisons between our four methods. Given that the HEAD method by definition does not return any content, it is excluded from this analysis. The remain-

Table 3 Number of DOIs resulting in 200-level responses across all four methods, distinguished by method and number of matching and non-matching URIs at the final link of their redirect chain

	HEAD		GET		GET+		Chrome	
	Match	!Match	Match	!Match	Match	!Match	Match	!Match
HEAD	X	X	4287	198	4360	125	4240	245
GET	4287	198	X	X	**4246**	239	**4126**	359
GET+	4360	125	4246	239	X	X	**4284**	201
Chrome	4240	245	4126	359	4284	201	X	X

ing three possible comparisons are: resources from the GET method compared to corresponding resources from the GET+ method, GET compared to Chrome, and GET+ compared to Chrome. Table 3 highlights the corresponding cells and their number of DOIs that are subject to our analysis.

There are many ways to compare the content of web resources. We follow the previously proven method of applying the Levenshtein distance and Cosine similarity measures to the resources' textual content [11]. Both measures provide different viewpoints on the notion of similarity. The Levenshtein distance is based on the amount of transactions needed to transform one string into the other and therefore focused on character similarity of textual content. Cosine, on the other hand, focuses more on the contextual similarity as it assesses commonality of salient terms in both compared strings. Used in combination, as previously done [11], both measures offer a comprehensive quantitative analysis of content similarity.

After dereferencing the DOIs and downloading the last link web resources, we needed to apply various post-processing methods in order to clean the text for a consistent comparison. First, we used the popular HTML parser BeautifulSoup[11] to strip all HTML markup and JavaScript. Second, we removed punctuation, stopwords (using the NLTK toolkit [12]), as well as excess white spaces and tabs. Lastly, for consistency, we converted all text to lowercase and removed accented characters.

Figure 6 shows the Cosine and Levenshtein values for all three comparisons. The DOIs are represented in the x-axis and the normalized similarity values on the y-axis. Cosine values, shown in red, are normalized so that a value of 1 represents the highest similarity and 0 greatest dissimilarity. The Levenshtein values, shown in green, are normalized the other way around. Figure 6a displays values for both similarity measures for the GET versus GET+ comparison, Fig. 6b for the GET versus Chrome, and Fig. 6c for the GET+ versus Chrome comparison. All three figures show a very similar picture with very high Cosine and very low Levenshtein values, both indicating very high levels of content similarity. By visual inspection, we can observe that Cosine values are slightly higher than Levenshtein values. Intuitively, that makes sense as Levenshtein registers minor editorial changes and Cosine focuses on the more general context. We also see somewhat better similarity scores for the GET versus GET+ comparison (Fig. 6a) than for the other two (Fig. 6b, c). A possible explanation for this small difference is the Chrome method, where JavaScript is executed that can result in the display of additional textual content. The GET and GET+ methods are command line based where JavaScript is submitted to the client, but it is not executed. However, it seems this additional content does not greatly affect the similarity.

These observations are confirmed by the mean, standard deviation, and median values of all comparisons and for both similarity measures presented in Table 4.

The findings offered in this section, summarized in Fig. 6 and Table 4, provide additional strong indicators that the resolution of DOIs is not consistent and, in fact, frequently leads to content that is dissimilar, as shown by a character and context comparison. These results therefore support our findings outlined in the previous sections that the end of the HTTP redirect chain, when resolving a DOI, depends on the request method used.

5 Future work

We see several directions for future work in this realm. We are highly motivated to compare our findings with other organizations, different network environments, and subscription levels. In addition, we are planning to deploy machine learning techniques to distinguish between resource types identified by our DOIs, with the goal to investigate whether some resource types (and corresponding web servers and services) respond more consistently to DOI referencing than others. Thirdly, we would like to further assess the impact of our findings, specifically the effects on web crawlers, archiving efforts, and various library systems. After all, we would argue that keeping infrastructure compliant with web standards will stimulate emerging research areas that rely not on human- but machine-assisted reading and machine learning.

Lastly, we would like to further engage with the persistent identifier community, including publishers and DOI registration agencies, to foster a dialogue about our findings, collaboratively seek solutions, and expand upon outreach

[11] https://www.crummy.com/software/BeautifulSoup/.

[12] https://www.nltk.org/.

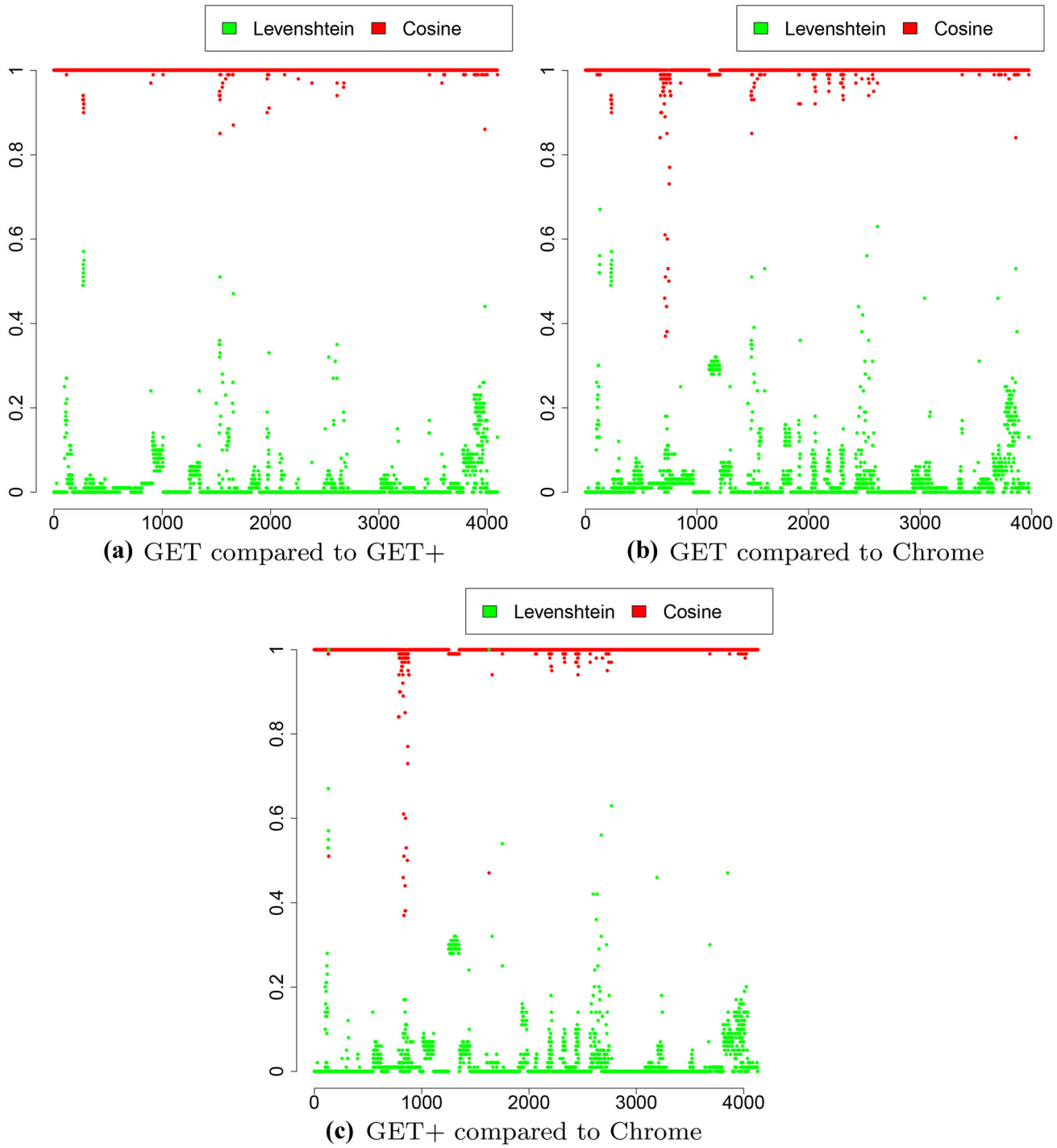


Fig. 6 Content similarity of dereferenced resources, compared by HTTP client

Table 4 Mean, standard deviation, and median values of Cosine and Levenshtein similarity measures for all three comparisons

	Cosine			Levenshtein		
	Mean	SD	Median	Mean	SD	Median
GET versus GET+	0.999	0.006	1	0.017	0.048	0
GET versus chrome	0.997	0.029	1	0.031	0.072	0.01
GET+ versus chrome	0.997	0.030	1	0.024	0.065	0

and training to ultimately better align the existing technical infrastructure and publishing environments with common web standards.

6 Conclusions

In this paper, we investigate the notion of persistence of DOIs as persistent identifiers from the perspective of their resolution on the web. Based on a previously generated corpus of DOIs and enhanced by an additional corpus of popular URIs, we present our results from dereferencing these resources with four very common but different HTTP request methods. We report on HTTP response codes, redirect chain length, and response code changes and highlight observed differences for requests originating from an external and internal network. We further analyze the effect of various access and licensing levels and offer a temporal analysis of response codes. In addition to the HTTP network level, we also conduct a preliminary study of content similarity of DOI-identified web resources.

We expected the resolution of DOIs to be consistent, but our findings do not show a consistent picture at all. More than half of all requests (51.7%) are unsuccessful from an external network compared to just over 33% from an institutional network. This number increased even further when repeating the experiment after some time. In addition, the success rate varies across request methods. We find that the method that most closely resembles the human browsing behavior (Chrome method) generally works best. We observed an alarming amount of changes in response code depending on the HTTP request method used. These findings provide strong indicators that scholarly content providers reply to DOI requests differently, depending on the request method, the originating network environment, and institutional subscription levels. Differences in textual content of dereferenced resources are small but noticeable and seem to further indicate variability in responses from content providers. Our scholarly record, to a large extent, relies on DOIs to persistently identify scholarly resources on the web. However, given our observed lack of consistency in DOI resolutions on the publishers' end, we raise serious concerns about the persistence of these persistent identifiers of the scholarly web.

References

1. Alam, S., Cartledge, C.L., Nelson, M.L.: Support for various HTTP methods on the web (2014). arXiv: 1405.2330
2. Bar-Yossef, Z., Broder, A.Z., Kumar, R., Tomkins, A.: Sic transit gloria telae: towards an understanding of the web's decay. In: Proceedings of WWW'04, pp. 328–337 (2004). https://doi.org/10.1145/988672.988716
3. Bilder, G.: January 2015 DOI outage: followup report (2015). https://www.crossref.org/blog/january-2015-doi-outage-followup-report/
4. Bordino, I., Boldi, P., Donato, D., Santini, M., Vigna, S.: Temporal evolution of the UK web. In: 2008 IEEE International Conference on Data Mining Workshops, pp. 909–918 (2008). https://doi.org/10.1109/ICDMW.2008.88
5. Brandman, O., Cho, J., Garcia-Molina, H., Shivakumar, N.: Crawler-friendly web servers. SIGMETRICS Perform. Eval. Rev. **28**(2), 9–14 (2000). https://doi.org/10.1145/362883.362894
6. Buriol, L.S., Castillo, C., Donato, D., Leonardi, S.: Temporal analysis of the wikigraph. In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings) (WI'06), vol. 06, pp. 45–51 (2006). https://doi.org/10.1109/WI.2006.164
7. Cho, J., Garcia-Molina, H.: The evolution of the web and implications for an incremental crawler. In: Proceedings of VLDB, vol. 00, pp. 200–209 (2000)
8. Cho, J., Garcia-Molina, H.: Estimating frequency of change. ACM Trans. Internet Technol. **3**, 256–290 (2003). https://doi.org/10.1145/857166.857170
9. Fielding, R.T., Reschke, J.: Hypertext transfer protocol (HTTP/1.1): semantics and content (2014). https://tools.ietf.org/html/rfc7231
10. Johnson, R., Watkinson, A., Mabe, M.: The STM report-an overview of scientific and scholarly publishing. International Association of Scientific, Technical and Medical Publishers (2018). https://www.stm-assoc.org/2018_10_04_STM_Report_2018.pdf
11. Jones, S.M., Van de Sompel, H., Shankar, H., Klein, M., Tobin, R., Grover, C.: Scholarly context adrift: three out of four URI references lead to changed content. PLoS ONE (2016). https://doi.org/10.1371/journal.pone.0167475
12. Klein, M.: 10,000 DOIs (2019). https://doi.org/10.6084/m9.figshare.7853462.v1
13. Klein, M., Balakireva, L.: On the persistence of persistent identifiers of the scholarly web. In: Proceedings of the 24th International Conference on Theory and Practice of Digital Libraries. TPDL, vol. 20, pp. 102–115 (2020). https://doi.org/10.1007/978-3-030-54956-5_8
14. Klein, M., Balakireva, L., Shankar, H.: Who is asking? Humans and machines experience a different scholarly web (2019). https://doi.org/10.17605/OSF.IO/SMCY2
15. Klein, M., Broadwell, P., Farb, S.E., Grappone, T.: Comparing published scientific journal articles to their pre-print versions. Int. J. Digital Lib. **20**, 335–350 (2019). https://doi.org/10.1007/s00799-018-0234-1
16. Klein, M., Sanderson, R., Van de Sompel, H., Warner, S., Haslhofer, B., Lagoze, C., Nelson, M.L.: A technical framework for resource synchronization. D-Lib Mag. **19**(1/2) (2013)
17. 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 (2014). https://doi.org/10.1371/journal.pone.0115253
18. Lawrence, S., Pennock, D.M., Flake, G.W., Krovetz, R., Coetzee, F.M., Glover, E., Nielsen, F.A., Kruger, A., Giles, C.L.: Persistence of web references in scientific research. Computer **34**(2), 26–31 (2001). https://doi.org/10.1109/2.901164
19. McCown, F., Chan, S., Nelson, M.L., Bollen, J.: The availability and persistence of web references in D-lib magazine (2005). arXiv: 0511077
20. Nelson, M.L., Smith, J..A., del Campo, I.G.: Efficient, automatic web resource harvesting. In: Proceedings of the 8th Annual ACM International Workshop on Web Information and Data Management. WIDM, vol. 06, pp. 43–50 (2006). https://doi.org/10.1145/1183550.1183560

21. Radinsky, K., Diaz, F., Dumais, S., Shokouhi, M., Dong, A., Chang, Y.: Temporal web dynamics and its application to information retrieval. In: Proceedings of the 6th ACM International Conference on Web Search and Data Mining. WSDM, vol. 13, pp. 781–782 (2013). https://doi.org/10.1145/2433396.2433500
22. Thompson, H.S., Tong, J.: Can common crawl reliably track persistent identifier (PID) use over time? (2018). arXiv: 1802.01424
23. Van de Sompel, H., Klein, M., Jones, S.M.: Persistent URIs must be used to be persistent. In: Proceedings of WWW, vol. 16, pp. 119–120 (2016). https://doi.org/10.1145/2872518.2889352