In 2020, I was a few months into my doctoral dissertation process with the idea that I would be researching plagiarism in the digital publication system. Here, I first encountered the true extent of link rot in digital scholarship. Link rot is the phenomenon of resources becoming inaccessible over time when their originally cited location is relocated or permanently unavailable.[1]

In my literature review, I would find articles that cited relevant resources to my topic, but the links to those artifacts were not available. A university research database usually provides access to most journals that I would like to read, but this was different. It was not that I was denied access; it was that access was no longer possible. The artifact had been lost to link rot, like many others before.

Research in the field of link rot has revealed some concerning statistics. Researchers at Harvard Law School’s Berkman Klein Center for Internet and Society found that around 50% of links in Supreme Court cases were broken.[2] Another study found that 70% of articles in the *Havard Law Review* were also broken.[1] While many early studies focused on the field of Law, studies on the broad collection of academic publications began to be conducted.

Library and Information scientists would point to persistent identifiers, such as DOIs, as a resolution to the problem. But how effective are DOIs? A study out of the Los Alamos National Laboratory looked at the availability of DOIs inside and outside an institution’s internal network. 33% of DOIs were inaccessible from within the internal network, and an astonishing 51.7% of DOIs were inaccessible outside the institution’s network. [3] In addition, a recent study from a Crossref employee admitted that 28% of the articles with DOIs appear to be entirely unpreserved. [4]

The deep dive into this topic engulfed me. As a result, I switched my dissertation topic to study the effects of link rot on scholarly publishing. I created a dataset of 2,500 articles, 100 from each year ranging from 2013 to 2022. The articles were also broken up into individual disciplines. Since past research often focused on Law, I decided to see if there were any exciting results between subject matters. I broke it up into five categories: Arts & Humanities, Business, Health & Medicine, STEM, and Social Sciences. 250 publications in their area represented each discipline.

My findings are closely aligned with previous research. The overall percentage of broken links was 36%, while the overall percentage of broken DOIs was 37%.[5] This, again, showed the ineffectiveness of our current DOI system. While there were no statistically significant findings regarding broken links across academic domains, there was between broken DOIs across them. Though the average percentage of broken DOIs was 37%, it ranged greatly from discipline to discipline. Business had an average broken DOI percentage of 44%, Arts & Humanities 42%, Social Sciences 40%, Health and Medicine 33%, and STEM 23%.[5] Further studies would need to be conducted to determine the cause of this disparity, but it does exist. There was anecdotal evidence that the chosen style guide for each field may have played a part.

To conduct my study, I was compelled to create a Python program that would scan PDFs for URIs, try to contact them, and record the HTTP response status code from the attempt. Python is a programming language that is often used to scrape and process data at a large scale. HTTP response status codes are records from the server of the website you are visiting, telling you how the status of the URI request that you send when you attempt to visit a link on the Internet. The program was also able to determine which URIs were DOIs or other common persistent identifiers such as arXiv links.

HTTP response status codes are separated into five classes or categories.[6]

* *1xx informational response* – the request was received, continuing process.
* *2xx successful* – the request was successfully received, understood, and accepted.
* *3xx redirection* – further action needs to be taken in order to complete the request.
* *4xx client error* – the request contains bad syntax or cannot be fulfilled.
* *5xx server error* – the server failed to fulfil an apparently valid request.

This Python program allowed me to process such a large dataset in a reasonable amount of time, though it still took time to process the results and run the proper analyses. Since the program was initially created, it has since expanded to be able to archive all active links using the Internet Archive. This program is open source and available at <https://github.com/rottingresearch/linkrot>.

While my Python program was able to help me complete my doctoral dissertation, there was still a lot of interest from the open-source community in this project. Unfortunately, not everyone has experience using Python and this made my program inaccessible to many. That is when the Rotting Research project was launched.

Seeing that there was a need in the community for a tool to check academic publications for the status of links that were cited, I began to create a web-based application that could be used by the general public. I found a great deal of assistance among the open-source community on GitHub. Many were able to contribute to different areas. Tim Robbins took on the task of designing the branding elements for our projects. Others like Aditi Rao were able to contribute to the code infrastructure. In total, we have over 2,000 contributions from 10 different countries. Together, we were able to create something special.

Rotting Research is the web application that was built out of these efforts. It is powered by the original Python program, which continues to be actively developed. Rotting Research has the same general functionality as the Python program. You upload a PDF and it begins to process the file, looking for links and sending HTTP requests to them all. One of the things that makes Rotting Research unique is that it then returns a generated report with a breakdown of the results.

The results are analyzed and returns a report that indicates how many DOIs are present, how many links are broken, as well as the error code that was returned from the server. You can view each URI that was tested as well as the response. Rotting Research also extracts any metadata from the PDF for the report. Finally, this report can be downloaded as a PDF for your records.

Rotting Research can be accessed at <https://rottingresearch.org/>.

Rotting Research isn’t just a tool. It also serves as a repository for information about the projects and current research in the field of link rot. There is a page dedicated to the latest research in this area of study. Another page is dedicated to the best practices for content creators to mitigate the risk of link rot in their work.

While Rotting Research serves a valuable role as is, it is far from resolving the issues that surround link rot in academic publishing. As a result, we will continue to develop the tool and add more functionality as efforts allow. We have a roadmap that has many exciting features.

We hope to add the ability to archive all active links via the Internet Archive soon. This feature is available in the Python program, but there are more considerations when uploading a large number of resources to the Internet Archive from a web server in bulk. We have found that, as of right now, the Internet Archive is the most reliable place to archive resources for long-term storage.

Another exciting feature that is coming soon to Rotting Research is a database of results that has been run through the web application. While this database will store the results of the test, there will be no other information stored. This database will allow our team, as well as other researchers, the ability to access a growing dataset of link rot results. We hope this will continue to fuel the impact that Rotting Research has on the academic community.

Down the road, we hope to address two other issues closely tied to link rot and data persistence in the academic record: retractions and content drift. It is no secret that articles are being retracted at a growing rate, with over 10,000 retractions from top journals in 2023. [7] We hope to integrate with a retraction database to notify users if an article that is cited has been retracted.

Change detection will allow us to address content drift by determining if a resource has changed its content since it was originally cited. This is a task that researchers Los Alamos National Laboratory are attempting to address with the Momento protocol. [8] This is a method in which a time stamp is integrated into a web resource to ensure that the version of an artifact that is cited is the one that is retrieved when navigated to.

Personally, I hope to continue my research into link rot and other issues with digital artifacts. Right now, I am investigating the potential impact that the Computer Science and Information Technology fields could have on improving the way that we access Internet Resources. The Computer Science community, as well as the technology industry, seems to have an issue with preserving systems, archives, and history.

Another area of research that I am focusing on is that of Internet protocols and how they can be improved to establish a more fortified infrastructure for the Internet. The HTTP status response code system appears to be showing signs that it needs revision. Backporting and backwards compatibility with rapidly developing standards could also help other industries who are unable to adapt as quickly as technology iterates.

Rotting Research has been lucky enough to receive so much support from the open-source community. We are currently a Docker-Sponsored Open-Source Software project. We also receive services as a member of the Red Hat Open-Source Infrastructure Program. Rotting Research has also been able to join the Open-Source Collective, a 501(c)(6) tax-exempt organization, and allows us accept donations. All of this, in addition to our generous contributors, we are excited to continue our mission of helping the academic community address and mitigate the effect of evanescence of our academic record.

Rotting Research is dedicated to its mission and encourages everyone to contribute code, ideas, or research to the field. To find out more, visit https://rottingresearch.org/contribute.

[1] K. Król and D. Zdonek, “Peculiarity of the bit rot and link rot phenomena,” *Glob. Knowl. Mem. Commun.*, vol. 69, no. 1/2, pp. 20–37, 2020, https://doi.org/10.1108/gkmc-06-2019-0067.

[2] J. L. Zittrain, K. Albert, and L. Lessig, “Perma: Scoping and Addressing the Problem of Link and Reference Rot in Legal Citations,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 2329161, Oct. 2013. https://doi.org/10.2139/ssrn.2329161.

[3] M. Klein and L. Balakireva, “An extended analysis of the persistence of persistent identifiers of the scholarly web,” *Int. J. Digit. Libr.*, Oct. 2021, https://doi.org/10.1007/s00799-021-00315-w.

[4] M. P. Eve, “Digital Scholarly Journals Are Poorly Preserved: A Study of 7 Million Articles,” *J. Librariansh. Sch. Commun.*, vol. 12, no. 1, Art. no. 1, Jan. 2024, https://doi.org/10.31274/jlsc.16288.

[5] M. A. Miller, “The Putrefaction of Digital Scholarship: How Link Rot Impacts the Integrity of Scholarly Publishing,” Jan. 2022, Accessed: Oct. 15, 2023. [Online]. Available: https://www.academia.edu/105022489/The\_Putrefaction\_of\_Digital\_Scholarship\_How\_Link\_Rot\_Impacts\_the\_Integrity\_of\_Scholarly\_Publishing

[6] “Hypertext Transfer Protocol (HTTP) Status Code Registry.” Accessed: Mar. 03, 2024. [Online]. Available: https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml

[7] R. Van Noorden, “More than 10,000 research papers were retracted in 2023 — a new record,” *Nature*, Dec. 2023, https://doi.org/10.1038/d41586-023-03974-8.

[8] M. Klein, H. Shankar, and H. Van de Sompel, “Robust Links in Scholarly Communication,” in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, Fort Worth Texas USA: ACM, May 2018, pp. 357–358. https://doi.org/10.1145/3197026.3203885.