Patching broken external links on Wikipedia

Harsha V. Madhyastha (University of Michigan)

1 Introduction

Crucial to any Wikipedia article, beyond its content, are the references and external links included in the article. These links to pages elsewhere on the web help the reader to not only understand and appreciate the article's content, but also to verify the claims presented in the article.

Problem: link rot. Unfortunately, even a few years after an external link is added to a Wikipedia article, the link often ceases to work. For example, consider the Wikipedia's article about Mars Express [6], a space exploration mission conducted by the European Space Agency. As seen in Figure 1, references 7 and 8 have been augmented with links to the Internet Archive's copies of the respective URLs. Whereas, reference 3 has been marked as permanently dead because no archived copy exists for it.


3. ^ Announcement by the European Space Agency on the launch of the *Mars Express* space probe: "Mars en route for the red planet". (2004). *Historic documents of 2003*. Washington, DC: CQ Press. Retrieved from http://library.cqpress.com/cqpac/hsdcp03p-229-9844-633819 [*permanent dead link*]
4. ^ "Mars Express: Summary". European Space Agency. March 29, 2011.
5. ^ "Mars Express". *NSSDC ID: 2003-022A*. NASA. Retrieved December 7, 2018.
6. ^ a b "Beagle 2 ESA/UK Commission of Inquiry". *NASASpaceFlight.com*. April 5, 2004. Retrieved March 29, 2016.
7. ^ "Glitch strikes Mars Express' radar boom - space - May 9, 2005 - New Scientist". Archived from the original on February 5, 2008.
8. ^ "Mars Express' kinky radar straightened out - space - May 12, 2005 - New Scientist". Archived from the original on February 6, 2008.
9. ^ a b c d "The spacecraft / Mars Express". ESA. October 10, 2005. Retrieved March 29, 2016.

Figure 1: **Examples of broken external links on Wikipedia.**

This problem of links on the web becoming dysfunctional – colloquially referred to as "link rot" – is not specific to the article discussed above; **even back in 2018, over 9 million references on Wikipedia included URLs that had become dysfunctional [8]**, and this number is growing over time [2]. These trends do not bode well for the preservation of Wikipedia for future generations. Wikipedia has been around for only a couple of decades – the blink of an eye in the timescale of human history – and, yet, so many links to relevant external information and services are already dysfunctional. This is particularly unfortunate given the tedious work that millions of users put in to ensure that articles on Wikipedia include appropriate citations to make them understandable and verifiable. Link rot is making a lot of this work go to waste.

Limitations of current solution. While link rot is not specific to Wikipedia, fixing the problem is easier on Wikipedia than elsewhere on the web, because anyone can edit any page on the site.

1

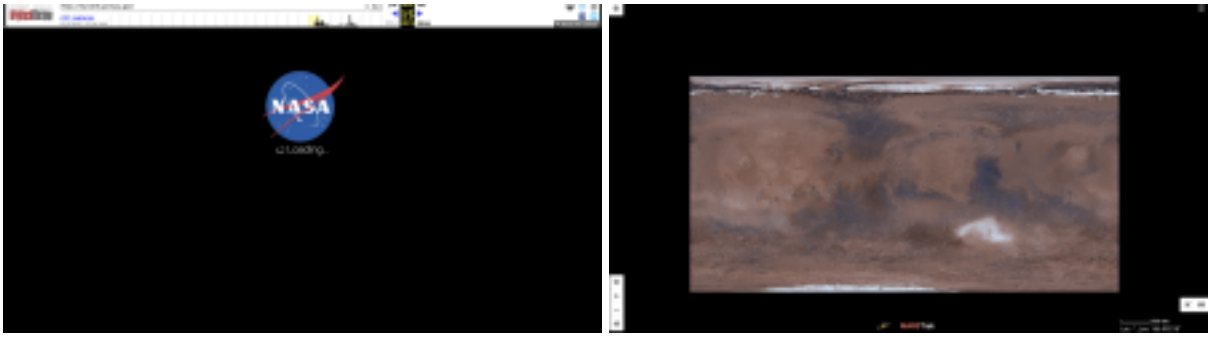
Problem with archived copy	**Affected article on Wikipedia**	**Observations**
Missing copy	Article [4] on Cy clamen persicum, a plant species	***Broken URL:*** http://apps.rhs.org.uk/plantselector/plant?plantid=5041 This is one of the many references in the article for which no archived copy exists; however, the linked page is still on the web ***New URL for same page:*** https://www.rhs.org.uk/plants/details?plantid=5041
Inaccessible service	Article [6] about European Space Agency's "Mars Express" mission	***Broken URL:*** http://marstrek.jpl.nasa.gov/ On this page, one can view and analyze imagery from Mars; this functionality does not work on archived copy [10] ***New URL for same page:*** https://trek.nasa.gov/mars/

Table 1: **Examples of problems associated with using archived copies from Internet Archive's Wayback Machine to cope with dysfunctional links seen on Wikipedia.**

Leveraging this capability, the InternetArchiveBot [5] continually scans Wikipedia articles to find broken external references; like in the example in Figure 1, it augments such references to include a link to an archived copy of the broken URL.

However, the approach of relying on archived page copies to patch broken links has two funda mental shortcomings; Table 1 lists an example for either case.

- First, since it is impractical to archive the entire web, **no archived copy exists for many broken URLs**. For example, on the English Wikipedia, there are close to 200,000 articles with such links, which are tagged as *"permanently dead"* [3].
- Second, **any functionality on a page which requires interactions with back-end servers does not work on archived copies**. For example, one of the broken external links [9] on the Mars Express Wiki article is to a page on which users can view and analyze imagery from Mars [7]; on the archived copy of that page [10], loading the image data spins forever (Figure 2).



(a) (b)

Figure 2: **Screenshots of (a) archived copy of http://marstrek.jpl.nasa.gov/ [10], and (b) live web page at https://trek.nasa.gov/mars/, where the same page now resides.**

Our vision. To sidestep these problems, we observe that many URLs cease to work only because the sites hosting them have been reorganized, and the URLs for pages on those sites have changed; this is the case for both the examples in Table 1. For such dysfunctional URLs, an archived copy of the page should not always be used as a *substitute* for the live page. Instead, archived information should be used to *locate* the page on the web and discover the page's new URL.

To automate this capability, we have been developing FABLE (Finding Aliases for Broken Links Efficiently). Given a broken URL, FABLE attempts to find an *alias* for it, i.e., the new URL for the page which was previously available at the now dysfunctional URL. Note that, given a broken link, our goal is not to find *alternate pages* with similar content. Instead, we seek to find if the specific page which a URL was previously pointing to still exists at an *alternate URL*, honoring the choice made in the past to link to this particular page.

In this one year project, our goals are two-fold. First, we aim to devise and implement the algo rithms necessary to efficiently discover a broken URL's alias (whenever one exists), while maxi mizing the accuracy of identified aliases. Second, we will apply these algorithms on links that have been marked as permanently dead by the InternetArchiveBot on Wikipedia [3]; these are broken links for which no archived copy exists on the Wayback Machine. We have been in conversations with Mark Graham, the Director of Wayback Machine at the Internet Archive, who oversees the team responsible for this bot. Mark has been enthusiastic about having the InternetArchiveBot use, in the future, the URL replacement patterns identified by FABLE to

replace broken links with their corresponding aliases.

Relevance to Wikimedia Foundation. One of the three main thrusts in Wikimedia's 2030 Strategic Direction is to improve the integrity of knowledge available on Wikipedia. All of the links to external web pages that millions of contributors and community editors have added to Wikipedia articles represent a key part of this effort. By addressing the problem of links becoming dysfunctional over time, our work aims to preserve the fruits of this effort for future generations. Moreover, by enabling broken links to be patched such that they continue to point to the pages they originally linked to, our work will ensure that users can access the latest content and all of the functionality available at those links, in contrast to relying on archived copies of pages.

2 Preliminary Work

Our current prototype of FABLE attempts to find the functional alias for any particular broken URL by combining the use of web archives and web search (Figure 3). We use the page title and content from Wayback Machine's last 200 status code archived copy of the broken URL to issue search queries on Google and Bing, restricting results to the URL's site. To account for changes to pages over time, we consider a search result to be a match if its title/content is *similar*, not necessarily *identical*, to that found in the archived copy. We compute the TF-IDF (term frequency-inverse document frequency) [14] similarity to compare pages and empirically tune the minimum similarity value necessary to be considered a match. We declare a matching search result as the broken URL's alias if it is the only one among the top 10 search results

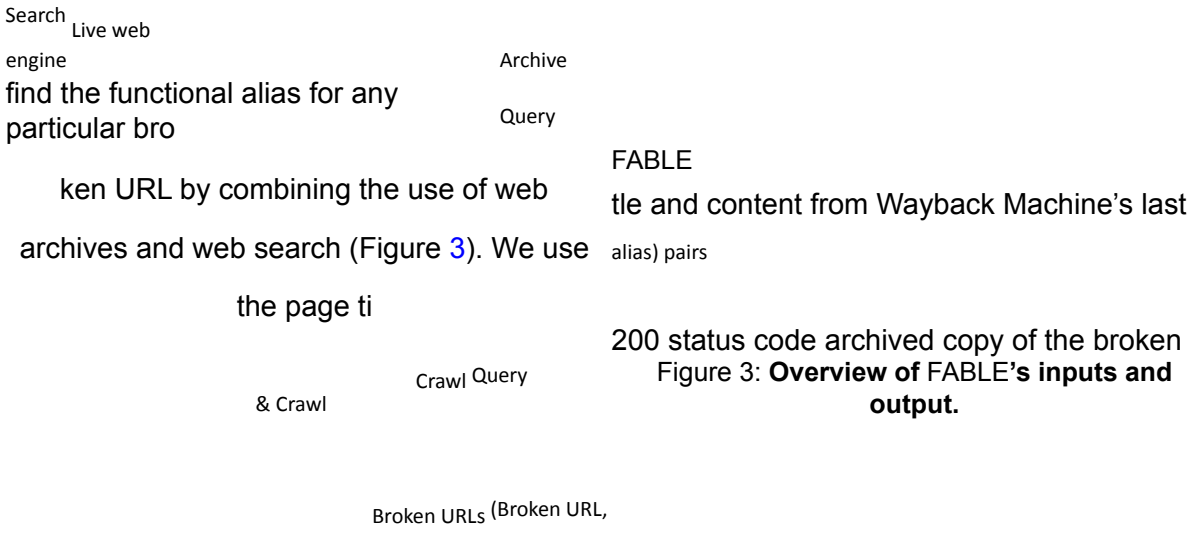


Figure 3: **Overview of** FABLE**'s inputs and output.**

whose similarity with the URL's archived copy is above our threshold.

This approach, however, fails to always correctly locate a broken URL's alias for several reasons.

- Even the most popular web archive, the Internet Archive, is incomplete. Thus, there might be no archived content for the broken URL, precluding the formulation of search queries. • Search results often do not include a URL's alias, even if it exists: 1) indices maintained by even popular web search engines are far from complete [15], 2) the latest archived content for the page's old URL may significantly differ from what is currently on the page, and 3) many pages have little textual content, making them not amenable for discovery via web search. • Simply comparing the archived content for the broken URL and the content that currently ex ists at any of the search results can end up matching a broken URL to the URL for an alternate page, which happens to have similar content.

To demonstrate these limitations of relying on web search to discover aliases, we applied our current prototype of FABLE on 1000 broken URLs for which we have ground truth information. This dataset comprises 500 *Alias* URLs whose aliases are known (based on the URL they previously redirected to) and 500 *NoAlias* URLs corresponding to pages which have been deleted (requests to

these URLs now return a HTTP 410 'Gone' response [1]); we manually vetted the 500 *Alias* URLs to confirm that their historical redirections were not erroneous.

Out of the 500 *Alias* URLs, we find the alias only for 37%. The aliases we find are also inaccurate in two ways: 1) we match 2.6% of the *Alias* URLs to a URL other than their alias, and 2) we find an alias for 1.6% of the *NoAlias* URLs, even though the pages previously available at these URLs no longer exist. Even if these limitations in coverage and accuracy did not exist, it is impractical to issue search queries for every broken URL and fetch the pages listed in the search results.

3 Proposed Work

In this project, we propose to refine FABLE to address the drawbacks of search-based discovery of aliases with respect to coverage, accuracy, and efficiency. We plan to focus specifically on

finding aliases for links which have been marked as *"permanently dead"* [3]. Using archived content to search the web is particularly inadequate for these links, since no archived copy exists from which a search query could be formulated.

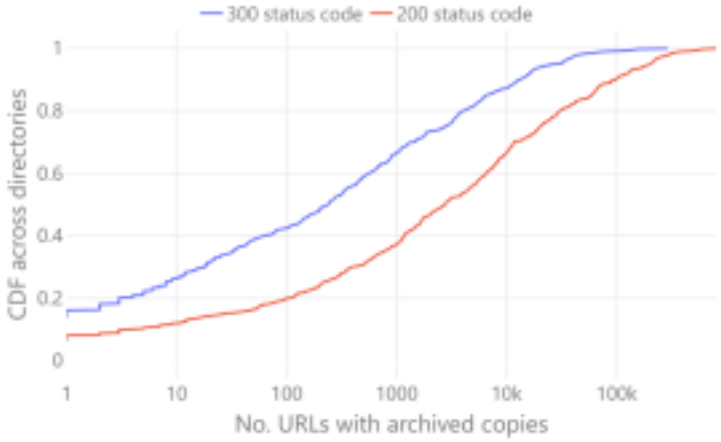


Figure 4: **Across 500 directories in which permanently dead links exist, cumulative distribution function (CDF) of the number of unique URLs for which the Wayback Machine has 200 status code and 3xx status code archived copies. A point at (x, y) means y% of directories have at most x URLs with archived copies.**

Our high-level insight is that, even when no archived copies exist on the Wayback Machine for a par ticular URL, it often has archived copies for other URLs in the same directory; the directory of a link is its prefix until the last '/' in its URL, e.g., https://processing.org/examples/flocking.html is in the di rectory *processing.org/examples*. Fig ure 4 shows this property across 500 directories that contain permanently dead links; these 500 directories are selected at random from a dataset of over 700,000 permanently dead links that Mark Graham at the Internet Archive has shared with us. By definition, a permanently dead link is one for which no 200 status code archived copy exists. However, as we see in the graph, for the median directory containing such

links, the Wayback Machine does have 200 status code archived copies for over a 1000 URLs and 300 status code archived copies for more than 100 URLs. We plan to exploit the presence of archived copies for a permanently dead link's neighbors (i.e., other URLs in the same directory) in several ways.

3.1 Historical redirections

First, though permanent dead links do not have any 200 status code archived copies, archived

copies with a 3xx status code exist for some of these links. Some of these correspond to historical redirections to the page's new URL which is functional today. These are cases where, after reorga nization, a site was previously redirecting requests for any page's old URL to the corresponding new URL, but the site subsequently lost the state necessary to perform these redirections.

For example, the URL http://www.natureserve.org/explorer/servlet/NatureServe?searchName=Geum+ peckii returns a "The requested page could not be found" error today. Since there are no 200 status code archived copies for this URL on the Wayback Machine, InternetArchiveBot has marked this link as permanently dead in the first reference of the article https://en.wikipedia.org/wiki/Geum peckii. But, this URL does have 301 status code archived copies (e.g., https://web.archive.org/web/20180630185757/http://www.natureserve.org/explorer/servlet/NatureServe?searchName=Geum+peckii), in which it redirects to http://explorer.natureserve.org/servlet/NatureServe?searchName=Geum+peckii. The latter link, which is the alias of the original link, works even today.

The challenge here is that some 3xx status code archived copies represent erroneous redirections, which are easy to identify in some cases (e.g., a redirection to the site's home page) but not always. **How to identify which redirections were erroneous and which ones were not?** To determine the correctness of redirections for any particular URL, we plan to leverage the typical presence of archived copies for the URL's neighbors. We will check whether that redirection is unique to that URL, or if the URL's neighbors also have archived copies with the same redirection. If multiple URLs were redirected to the same final URL, it indicates that the redirection is likely to be the default behavior for many URLs set by the site host, and thus, the final URL cannot be the alias. Note that this check must be done with care because the redirections on site.com might be unique only because any particular *u* may redirect to a URL which contains *u*, e.g., site.com/error.php?u.

3.2 Inference of URL replacement patterns

Taking advantage of historical redirections will help us find aliases for more broken links than possible only with web search. However, many issues remain. First, the fraction of broken links

for which we can find aliases by relying on historical redirections heavily depends on which links have any 3xx status code archived copies. Second, while checking the archived copies of a broken link is more efficient than web search based alias discovery, querying Wayback Machine for hun dreds of thousands of dead links takes a lot of time. Lastly, we still need to address the fact that some of the aliases discovered via web search may be inaccurate.

To address these problems with respect to coverage, efficiency, and accuracy, we plan to exploit the property that, when a page's URL changes, it is usually the case that URLs of other similar pages (e.g., those in the same directory) also change. This is because the changes in page URLs are typically the result of the reorganization of an entire site or subdomain; it is rarely the case that the URL of a single page has been changed. Consequently, there is usually a pattern in how the old URLs for pages on a site map to their corresponding new URLs (i.e., their aliases). We plan to leverage these patterns in three ways.

Better coverage. First, even if no archived copies exist for a particular broken URL, we can try to find aliases for the URL's neighbors and use the pattern seen in those to infer the URL's alias. For example, on the article https://en.wikipedia.org/wiki/Eastern moa, the reference to http://www. taxonomy.nl/Main/Classification/51296.htm has been marked as permanently dead because all of the Wayback Machine's archived copies of this URL are erroneous. However, Wayback Machine does have functional copies for other URLs in the same directory, such as http://www.taxonomy.nl: 80/Main/Classification/1002896.htm and http://www.taxonomy.nl:80/Main/Classification/1002868. htm. We can use web search to discover the aliases for these URLs – http://taxonomicon.taxonomy. nl/TaxonLinks.aspx?id=1002896 for the former and http://taxonomicon.taxonomy.nl/TaxonLinks. aspx?id=1002868 for the letter – and thereby infer that the alias for the permanent dead link is http://taxonomicon.taxonomy.nl/TaxonTree.aspx?id=51296.

We have begun experimenting with Microsoft Excel's Flash Fill [11] to enable such inference of aliases. Flash Fill is an instance of the programming by example (PBE) [13] approach wherein,

given a number of input to output examples, the output can be predicted for a new input.

The primary research question here is: **what inputs are necessary to enable such inference**

of aliases for broken URLs? It is often infeasible to infer the alias for a broken link simply based on the URL. For example, based on historical redirections for http://lvrj.com/news/16976731.html (a link which does not work today), we know that the same page is now at http://reviewjournal.com/ news/patients-vent-emotions. If no archived copies existed for this broken link, then we would still need the page's title in order to infer the page's new URL. In such cases, we plan to extract page titles for permanently dead links from the references section of Wikipedia articles.

Better efficiency. Second, due to the patterns in how URLs are transformed when a site is reorga nized, our system FABLE needs to use web search or historical redirections to discover the aliases for only a few URLs in each directory. Thereafter, the patterns inferred from these aliases can be leveraged to infer the new URLs for other broken URLs in the same directory, without having to query web search engines or crawl pages from the live web or from Wayback Machine.

We will investigate **what is the format in which URL replacement patterns for any particular site should be represented?** This is important so that, in the future, the InternetArchiveBot can use the URL replacement rules output by FABLE to patch broken links on Wikipedia with their corresponding aliases. Whenever it finds a broken link, the format of our rules should enable the InternetArchiveBot to independently generate that link's alias (if one exists) by applying one of the rules that we have published for that link's site.

Better accuracy. Lastly, we will exploit the patterns that exist between old and new URLs to confirm the accuracy of any alias we identify. For example, if we find the alias for a particular URL using web search or by examining historical redirections, we will consider the alias to be correct only if the relationship between this URL and its alias is in keeping with aliases found for other broken URLs in the same site. This will also help us account for inaccuracy in the inputs provided for inferring URL replacement patterns.

The question that we seek to address here is: **for any target broken URL, what is the minimum number of neighboring URLs for which we need to identify aliases in order to be able to infer the target's alias accurately?** On the one hand, finding aliases for only a few neighbors might

not suffice to see the underlying pattern in how old URLs on the site got transformed to the cor responding new URLs. On the other hand, trying to find aliases for more neighbors implies we need to spend more CPU and network bandwidth in issuing search queries, crawling search re sults, looking up archived copies of pages, etc.

4 Expected Outcomes and Execution Plan

We envision three primary outcomes from this one-year project.

- First, our work will result in a set of algorithms which, given a previously functional URL that does not work today, will identify the new URL at which the same page now exists. Our algorithms will aim to maximize coverage (i.e., successfully discover the new URLs for pages that still exist), accuracy (i.e., ensure that identified aliases are correct), and efficiency (i.e., minimize the amount of CPU/network bandwidth spent in finding aliases).
- We will implement our algorithms in our FABLE system, which will be extensible to find aliases for dead links that appear anywhere on the web. We will host FABLE's source code in a publicly available GitHub repository, so that others can build on our work in the future. But, to ensure the sustainability of our work, the primary utility of our system will be the URL replacement patterns that we discover from running FABLE on links which the Inter netArchiveBot has tagged as permanently dead [3]. Over the past year, we have interacted fre quently with Mark Graham, who is the Director of Wayback Machine at the Internet Archive. He has expressed interest in having the InternetArchiveBot use the URL replacement patterns identified by FABLE. Modifying the InternetArchiveBot to use this new input is beyond the one-year scope of this project.
- Lastly, we will aim to publish a research paper summarizing our work and our findings. This will help others both reproduce and extend our work.

To produce these outcomes, we anticipate executing our proposed work in three phases.

Months 1–4: Improve coverage. In the first phase, we will implement our proposed approaches – using historical redirections and via inference based on URL replacement patterns – for finding

aliases for more broken links than we can today. We will apply our methods on permanent dead links identified by the InternetArchiveBot [3] as this offers two advantages: 1) the lack of archived

copies for these links means that the inadequacy of existing methods for rescuing them is more acute, and 2) since these links have already been marked as permanently dead, it saves us the effort of checking which links are broken and have no archived copies on the Wayback Machine. We will run our system, FABLE, on the Microsoft Azure cloud service because our current prototype is implemented to work on that platform. Paying for the graduate student time necessary to port our implementation to work on Wikimedia infrastructure will cost more than the $500 that our budget currently includes for running FABLE on Azure.

Months 5–8: Improve accuracy. Our second milestone will be to use inferred URL replacement patterns to maximize the fraction of the aliases we find that are correct; our methods to discover the new URLs for dead links are useful only if the vast majority of aliases we find are correct. To evaluate how we fare on this front, we will conduct the following user study. For a random sample of the aliases we find, we plan to make a post on the 'Talk' pages of the corresponding articles soliciting feedback from the editors of those articles. The post we make for any particular alias will provide all the information that FABLE used to find this alias. In order to make these posts, we will go through the standard bot approval process [12] and seek community input on how best to run such a study. We have previously had conversations with Diego-Saez-Trumper, a ´ Senior Research Scientist at Wikimedia Research, about this.

Based on the results from our study, we will work on improving the accuracy of our methods. After refining our implementation, we will rerun our user survey on another random sample of links to measure the improvement.

Months 9–12: Improve efficiency and publish outcomes. In the last phase, we will focus on ensuring the sustainability of our work. For this, we will first focus on determining the format in which the URL replacement patterns discovered by FABLE should be represented, and how to minimize the amount of work the FABLE has to do to infer these patterns. We will also consult with the Internet Archive, so that the URL replacement patterns we publish can be used in the operation of the InternetArchiveBot in the future.

In this final phase of the project, we will also write up a paper which will describe FABLE's algorithms and implementation, as well as our findings from running the system. We will submit the paper for publication to a top-tier research conference.

References

[1] 410 gone - HTTP. https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/410.

[2] Category:Articles with dead external links - Wikipedia. https://en.wikipedia.org/wiki/Category:Articles with dead external links.

[3] Category:articles with permanently dead external links. https://en.wikipedia.org/wiki/Category:Articles with permanently dead external links.

[4] Cyclamen persicum - Wikipedia. https://en.wikipedia.org/wiki/Cyclamen persicum.

[5] InternetArchiveBot. https://meta.wikimedia.org/wiki/InternetArchiveBot. [6] Mars Express - Wikipedia. https://en.wikipedia.org/wiki/Mars Express. [7] Mars Trek. https://trek.nasa.gov/mars/.

[8] More than 9 million broken links on Wikipedia are now rescued. https://blog.archive.org/2018/10/01/more-than-9-million-broken-links-on-wikipedia-are-now-rescued/.

[9] NASA Mars Trek. http://marstrek.jpl.nasa.gov/.

[10] NASA Mars Trek (copy on August 15, 2015). https://web.archive.org/web/20150815084859/http://marstrek.jpl.nasa.gov/.

[11] Using Flash Fill in Excel. https://support.microsoft.com/en-us/office/using-flash-fill-in-excel 3f9bcf1e-db93-4890-94a0-1578341f73f7.

[12] Wikipedia:bots/requests for approval. https://en.wikipedia.org/wiki/Wikipedia: Bots/Requests for approval.

[13] D. C. Halbert. *Programming by example*. PhD thesis, University of California, Berkeley, 1984.

[14] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[15] A. Van den Bosch, T. Bogers, and M. De Kunder. Estimating search engine index size vari ability: A 9-year longitudinal study. *Scientometrics*, 107(2):839–856, 2016.