WikiHist.html: English Wikipedia's Full Revision History in HTML Format

Blagoj Mitrevski*
EPFL
blagoj.mitrevski@epfl.ch

Tiziano Piccardi*
EPFL
tiziano.piccardi@epfl.ch

Robert West
EPFL
robert.west@epfl.ch

Abstract

Wikipedia is written in the wikitext markup language. When serving content, the MediaWiki software that powers Wikipedia parses wikitext to HTML, thereby inserting additional content by expanding macros (templates and modules). Hence, researchers who intend to analyze Wikipedia as seen by its readers should work with HTML, rather than wikitext. Since Wikipedia's revision history is publicly available exclusively in wikitext format, researchers have had to produce HTML themselves, typically by using Wikipedia's REST API for ad-hoc wikitext-to-HTML parsing. This approach, however, (1) does not scale to very large amounts of data and (2) does not correctly expand macros in historical article revisions. We solve these problems by developing a parallelized architecture for parsing massive amounts of wikitext using local instances of MediaWiki, enhanced with the capacity of correct historical macro expansion. By deploying our system, we produce and release WikiHist.html, English Wikipedia's full revision history in HTML format. We highlight the advantages of WikiHist.html over raw wikitext in an empirical analysis of Wikipedia's hyperlinks, showing that over half of the wiki links present in HTML are missing from raw wikitext, and that the missing links are important for user navigation. Data and code are publicly available at https://doi.org/10.5281/zenodo.3605388.

1 Introduction

Wikipedia constitutes a dataset of primary importance for researchers across numerous subfields of the computational and social sciences, such as social network analysis, artificial intelligence, linguistics, natural language processing, social psychology, education, anthropology, political science, human–computer interaction, and cognitive science. Among other reasons, this is due to Wikipedia's size, its rich encyclopedic content, its collaborative, self-organizing community of volunteers, and its free availability.

Anyone can edit articles on Wikipedia, and every edit results in a new, distinct revision being stored in the respective article's history. All historical revisions remain accessible via the article's *View history* tab.

*Authors contributed equally.

Wikitext and HTML. Wikipedia is implemented as an instance of MediaWiki,[1] a content management system written in PHP, built around a backend database that stores all information. The content of articles is written and stored in a markup language called *wikitext* (also known as *wiki markup* or *wikicode*).[2] When an article is requested from Wikipedia's servers by a client, such as a Web browser or the Wikipedia mobile app, MediaWiki translates the article's wikitext source into HTML code that can be displayed by the client. The process of translating wikitext to HTML is referred to as *parsing*. An example is given below, in Fig. 1.

Wikitext: '''Niue''' ({{lang-niu|Niuē}}) is an [[island country]].

HTML: Niue (Niuean: Niuē) is an island country.

Figure 1: Example of wikitext parsed to HTML.

Wikitext provides concise constructs for formatting text (e.g., as bold, cf. yellow span in the example of Fig. 1), inserting hyperlinks (cf. blue span), tables, lists, images, etc.

Templates and modules. One of the most powerful features of wikitext is the ability to define and invoke so-called *templates*. Templates are macros that are defined once (as wikitext snippets in wiki pages of their own), and when an article that invokes a template is parsed to HTML, the template is expanded, which can result in complex portions of HTML being inserted in the output. For instance, the template *lang-niu,* which can be used to mark text in the Niuean language, is defined in the Wikipedia page *Template:lang-niu*, and an example of its usage is marked by the red span in the example of Fig. 1. Among many other things, the infoboxes appearing on the top right of many articles are also produced by templates. Another kind of wikitext macro is called *module.* Modules are used in a way similar to templates, but are defined by code in the Lua programming language, rather than wikitext.

[1] https://www.mediawiki.org
[2] https://en.wikipedia.org/wiki/Help:Wikitext

Researchers' need for HTML. The presence of templates and modules means that the HTML version of a Wikipedia article typically contains more, oftentimes substantially more, information than the original wikitext source from which the HTML output was produced. For certain kinds of study, this may be acceptable; e.g., when researchers of natural language processing use Wikipedia to train language models, all they need is a large representative text corpus, no matter whether it corresponds to Wikipedia as seen by readers. On the contrary, researchers who study the very question how Wikipedia is consumed by readers cannot rely on wikitext alone. Studying wikitext instead of HTML would be to study something that regular users never saw.

Unfortunately, the official Wikipedia dumps provided by the Wikimedia Foundation contain wikitext only, which has profound implications for the research community: researchers working with the official dumps study a representation of Wikipedia that differs from what is seen by readers. To study what is actually seen by readers, one must study the HTML that is served by Wikipedia. And to study what was seen by readers in the past, one must study the HTML corresponding to historical revisions. Consequently, it is common among researchers of Wikipedia (Dimitrov et al. 2017; Lemmerich et al. 2019; Singer et al. 2017) to produce the HTML versions of Wikipedia articles by passing wikitext from the official dumps to the Wikipedia REST API,[3] which offers an endpoint for wikitext-to-HTML parsing.

Challenges. This practice faces two main challenges:

1. Processing time: Parsing even a single snapshot of full English Wikipedia from wikitext to HTML via the Wikipedia API takes about 5 days at maximum speed. Parsing the full history of all revisions (which would, e.g., be required for studying the evolution of Wikipedia) is beyond reach using this approach.
2. Accuracy: MediaWiki (the basis of the Wikipedia API) does not allow for generating the exact HTML of historical article revisions, as it always uses the latest versions of all templates and modules, rather than the versions that were in place in the past. If a template was modified (which happens frequently) between the time of an article revision and the time the API is invoked, the resulting HTML will be different from what readers actually saw.

Given these difficulties, it is not surprising that the research community has frequently requested an HTML version of Wikipedia's dumps from the Wikimedia Foundation.[4]

Dataset release: WikiHist.html. With the WikiHist.html dataset introduced in this paper, we address this longstanding need and surmount the two aforementioned hurdles by releasing the complete revision history of English Wikipedia in HTML format. We tackle the challenge of scale (challenge 1 above) by devising a highly optimized, parallel data processing pipeline that leverages locally installed MediaWiki instances, rather than the remote Wikipedia API, to parse nearly 1 TB (bzip2-compressed) of historical wikitext, yielding about 7 TB (gzip-compressed) of HTML.

We also solve the issue of inconsistent templates and modules (challenge 2 above) by amending the default MediaWiki implementation with custom code that uses templates and modules in the exact versions that were active at the time of the article revisions in which they were invoked. This way, we approximate what an article looked like at any given time more closely than what is possible even with the official Wikipedia API.

In addition to the data, we release a set of tools for facilitating bulk-downloading of the data and retrieving revisions for specific articles.

Download location. Both data and code can be accessed via https://doi.org/10.5281/zenodo.3605388.

Paper structure. In the remainder of this paper, we first describe the WikiHist.html dataset (Sec. 2) and then sketch the system we implemented for producing the data (Sec. 3). Next, we provide strong empirical reasons for using WikiHist.html instead of raw wikitext (Sec. 4), by showing that over 50% of all links among Wikipedia articles are not present in wikitext but appear only when wikitext is parsed to HTML, and that these HTML-only links play an important role for user navigation, with click frequencies that are on average as high as those of links that also appear in wikitext before parsing to HTML.

2 Dataset description

The WikiHist.html dataset comprises three parts: the bulk of the data consists of English Wikipedia's full revision history parsed to HTML (Sec. 2.1), which is complemented by two tables that can aid researchers in their analyses, namely a table of the creation dates of all articles (Sec. 2.2) and a table that allows for resolving redirects for any point in time (Sec. 2.3). All three parts were generated from English Wikipedia's revision history in wikitext format in the version of 1 March 2019. For reproducibility, we archive a copy of the wikitext input[5] alongside the HTML output.

2.1 HTML revision history

The main part of the dataset comprises the HTML content of 580M revisions of 5.8M articles generated from the full English Wikipedia history spanning 18 years from 1 January 2001 to 1 March 2019. Boilerplate content such as page headers, footers, and navigation sidebars are not included in the HTML. The dataset is 7 TB in size (gzip-compressed).

Directory structure. The wikitext revision history that we parsed to HTML consists of 558 bzip2-compressed XML files, with naming pattern enwiki-20190301-pages-meta-history$1.xml-p$2p$3.bz2, where $1 ranges from 1 to 27, and p$2p$3 indicates that the file contains revisions for pages with ids between $2 and $3. Our dataset mirrors this structure and contains one directory per original XML file, with the same name. Each directory contains a collection of gzip-compressed JSON files, each containing 1,000 HTML

[3] https://en.wikipedia.org/w/api.php

[4] See, e.g., https://phabricator.wikimedia.org/T182351.

[5] Downloaded from https://dumps.wikimedia.org/enwiki/.

article revisions. Since each original XML file contains on average 1.1M article revisions, there are around 1,100 JSON files in each of the 558 directories.

File format. Each row in the gzipped JSON files represents one article revision. Rows are sorted by page id, and revisions of the same page are sorted by revision id. As in the original wikitext dump, each article revision is stored in full, not merely as a diff from the previous revision. In order to make WikiHist.html a standalone dataset, we include all revision information from the original wikitext dump, the only difference being that we replace the revision's wikitext content with its parsed HTML version (and that we store the data in JSON rather than XML).

The schema therefore mirrors that of the original wikitext XML dumps, [6] but for completeness we also summarize it in Table 1a.

Hyperlinks. In live Wikipedia, hyperlinks between articles appear either as blue or as red. Blue links point to articles that already exist (e.g., /wiki/Niue), whereas red links indicate that the target article does not exist yet (e.g., /w/index.php?title=Brdlbrmpft&action=edit&redlink=1). This distinction is not made in the wikitext source, where all links appear in identical format (e.g., [[Niue]], [[Brdlbrmpft]]), but only when the respective article is requested by a client and parsed to HTML. As the existence of articles changes with time, we decided to not distinguish between blue and red links in the raw data and render all links as red by default. In order to enable researchers to determine, for a specific point in time, whether a link appeared as blue or red and what the hyperlink network looked like at that time, we also provide the two complementary datasets described next.

2.2 Page creation times

The lookup file page_creation_times.json.gz (schema in Table 1b) specifies the creation time of each English Wikipedia page. To determine if a link to a target article A was blue or red at time t (cf. Sec. 2.1), it suffices to look up A in this file. If A was created after time t or if it does not appear in the file, the link was red at time t; otherwise it was blue.

2.3 Redirect history

Wikipedia contains numerous redirects, i.e., pages without any content of their own whose sole purpose is to forward traffic to a synonymous page. For instance, *Niue Island* redirects to *Niue*. Link occurrences in the wikitext dumps, as well as our derived HTML dumps, do not specify whether they point to a proper article or to a redirect. Rather, redirects need to be explicitly resolved by researchers themselves, a step that is complicated by the fact that redirect targets may change over time. Since redirect resolution is crucial for analyzing Wikipedia's hyperlink network, we facilitate this step by also releasing the full redirect history as a supplementary dataset: the file redirect_history.json.gz (schema in Table 1c) specifies all revisions corresponding to redirects, as well as the target page to which the respective page redirected at the time of the revision.

[6] https://www.mediawiki.org/w/index.php?title=Help:Export&oldid=3495724#Export_format

Table 1: JSON schemas of dataset. All fields in HTML revision history are copied from wikitext dump, except html, which replaces the original text.

(a) HTML revision history (Sec. 2.1)

Field name	*Description*
id	id of this revision
parentid	id of revision modified by this revision
timestamp	time when revision was made
cont_username	username of contributor
cont_id	id of contributor
cont_ip	IP address of contributor
comment	comment made by contributor
model	content model (usually wikitext)
format	content format (usually text/x-wiki)
sha1	SHA-1 hash
title	page title
ns	namespace (always 0)
page_id	page id
redirect_title	if page is redirect, title of target page
html	revision content in HTML format

(b) Page creation times (Sec. 2.2)

Field name	*Description*
page_id	page id
title	page title
ns	namespace (0 for articles)
timestamp	time when page was created

(c) Redirect history (Sec. 2.3)

Field name	*Description*
page_id	page id of redirect source
title	page title of redirect source
ns	namespace (0 for articles)
revision_id	revision id of redirect source
timestamp	time at which redirect became active
redirect	page title of redirect target (in 1st item of array; 2nd item can be ignored)

2.4 Limitation: deleted pages, templates, modules

Wikipedia's wikitext dump contains all historical revisions of all pages that still existed at the time the dump was created. It does not, however, contain any information on pages that were deleted before the dump was created. In other words, when a page is deleted, its entire history is purged. Therefore, since WikiHist.html is derived from a wikitext dump, deleted pages are not included in WikiHist.html either.

When using WikiHist.html to reconstruct a past state of Wikipedia, this can lead to subtle inaccuracies. For instance, it follows that the rule of Sec. 2.2 for deciding whether a link was blue or red at time t will incorrectly tag a link (u, v) as red if v existed at time t but was deleted before 1 March 2019 (the date of the wikitext dump that we used). Although such

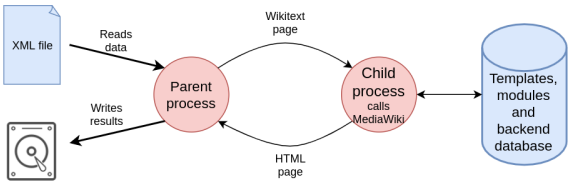

Figure 2: Architecture for parsing Wikipedia's revision history from wikitext to HTML.

inconsistencies are exceedingly rare in practice, researchers using WikiHist.html should be aware of them.

Since MediaWiki handles templates and Lua modules (together referred to as *macros* in the remainder of this section) the same way it treats articles (they are normal wiki pages, marked only by a prefix *Template:* or *Module:*), deleted macros are not available in the revision history either. It follows that a deleted macro cannot be processed, even when parsing a revision created at a time before the macro was deleted. This leads to unparsed wikitext remaining in the HTML output in the case of templates, and to error messages being inserted into the HTML output in the case of Lua modules.

In some cases, we observed that editors deleted a macro and created it again with the same name later. This action introduces the problem of losing the revision history of the macro before its second creation. In such cases, we assume that the oldest macro revision available approximates best how the macro looked before its deletion and use that version when parsing article revisions written before the macro was deleted.

We emphasize that the limitation of deleted pages, templates, and modules is not introduced by our parsing process. Rather, it is inherited from Wikipedia's deliberate policy of permanently deleting the entire history of deleted pages. Neither can the limitation be avoided by using the Wikipedia API to parse old wikitext revisions; the same inconsistencies and error messages would ensue. On the contrary, WikiHist.html produces strictly more accurate approximations of the HTML appearance of historical revisions than the Wikipedia API, for the API always uses the latest revision of all templates and modules, rather than the revision that was actually in use at the time of the article revision by which it was invoked.

3 System architecture and configuration

Wikipedia runs on MediaWiki, a content management system built around a backend database that stores all information on pages, revisions, users, templates, modules, etc. In this project we only require one core functionality: parsing article content from wikitext to HTML. In MediaWiki's intended use case, parsing is performed on demand, whenever a page is requested by a Web client. Our use case, on the contrary, consists in bulk-parsing a very large number of revisions. Since MediaWiki was not built for such bulk-parsing, the massive scale of our problem requires a carefully designed system architecture.

System overview. Our solution is schematically summarized in Fig. 2. As mentioned in Sec. 2.1, the input to the parsing process consists of the hundreds of XML files that make up English Wikipedia's full revision history in wikitext format. Our system processes the XML files in parallel, each in a separate parent process running on a CPU core of its own. Parent processes read the data from disk (in a streaming fashion using a SAX XML parser) and spawn child processes that parse the article contents from wikitext to HTML. Each child process has access to its own dedicated MediaWiki instance. The parent processes collect the HTML results from the child processes and write them back to disk. Although this architecture is straightforward in principle, several subtleties need to be handled, described next.

Template and module expansion. Wikitext frequently invokes macros (templates and modules) that need to be expanded when parsing to HTML. Since macros may (and frequently do) themselves change over time, it is important to use the version that was active at the time of the article revision that is being parsed, given that we aim to reconstruct the HTML as it appeared at the time of the article revision. MediaWiki unfortunately does not provide such a retroactive macro expansion mechanism, but instead always uses the latest available version of each macro. We therefore provide a workaround ourselves, by implementing an interceptor that, every time a macro is expanded, selects the historically correct macro version based on the revision date of the page being parsed, and returns that macro version to the parser instead of the default, most recent version.[7] More precisely, we select the most recent macro version that is older than the article revision being parsed.

MediaWiki version. Not only templates and modules, but also the MediaWiki software itself has changed over time, so in principle the same wikitext might have resulted in different HTML outputs at different times. To strictly reproduce the exact HTML served by Wikipedia at a given time, one would need to use the MediaWiki version deployed by Wikipedia at that time. Juggling multiple versions of MediaWiki would, however, severely complicate matters, so we started by consulting the Internet Archive Wayback Machine[8] in order to compare identical article revisions in different HTML snapshots taken at times between which live Wikipedia's MediaWiki version changed. Screening numerous revisions this way, we found no noticeable differences in the HTML produced by different MediaWiki versions and therefore conclude that it is safe to use one single MediaWiki version for all revisions. In particular, we use the latest long-term support version of MediaWiki, 1.31.[9]

Parser extensions. MediaWiki offers numerous extensions, but not all extensions used by live Wikipedia are preinstalled in MediaWiki's default configuration. We therefore manually installed all those extensions (including their dependencies) that are necessary to reproduce live Wikipe-

[7] To support this procedure, the caching mechanisms of MediaWiki must be turned off, which introduces significant latency.

[8] https://archive.org/web/

[9] https://www.mediawiki.org/wiki/MediaWiki_1.31

dia's parsing behavior. In particular, we mention two crucial parser extensions: *ParserFunctions*,[10] which allows for conditional clauses in wikitext, and *Scribunto*,[11] the extension that enables the usage of Lua modules in wikitext.

Database connectivity. By design, MediaWiki instances cannot run without a persistent database connection. However, given that (1) wikitext-to-HTML parsing is the only functionality we require, (2) the input to be parsed comes directly from a wikitext dump rather than the database, and (3) we intercept template and module lookups with custom code (see above), we never actually need to touch the MediaWiki database. Hence we need not populate the database with any data (but we still need to create empty dummy tables in order to prevent MediaWiki from throwing errors).

Scaling up. Given the amount of wikitext in the full revision history, parallelization is key when parsing it. We explored multiple common solutions for scaling up, including Spark and Yarn, but none of them satisfied all our requirements. Therefore, we instead settled on a custom, highly-optimized implementation based on Docker[12] containers: we bundle the modified MediaWiki installation alongside the required MySQL database into a standalone Docker container and ship it to each machine involved in the data processing.

Failure handling. Failures can happen during the parsing process for multiple reasons, including malformed wikitext, memory issues, etc. Detecting such failures is not easy in MediaWiki's PHP implementation: in case of an error it calls the `die` function, which in turn interrupts the process without raising an exception. As a workaround, the parent processes (one per XML file; see above) are also responsible for monitoring the status of the child processes: whenever one of them fails, the event is detected and logged. By using these logs, processing of the failure-causing revisions can be resumed later, after writing custom code for recognizing problematic wikitext and programmatically fixing it before sending it to the parser. Our deployed and released code incorporates all such fixes made during development runs.

Computation cost. We used 4 high-end servers with 48 cores and 256 GB of RAM each. Each core ran one parent and one child process at a time. In this setup, parsing English Wikipedia's full revision history from wikitext to HTML took 42 days and, at a price of CHF 8.70 per server per day, cost a total of CHF 1,462.

4 Advantages of HTML over wikitext

Our motivation for taking on the considerable effort of parsing Wikipedia's entire revision history from wikitext to HTML was that raw wikitext can only provide an approximation of the full information available in a Wikipedia article, primarily because the process of parsing wikitext to HTML tends to pull in information implicit in external templates and modules that are invoked by the wikitext.

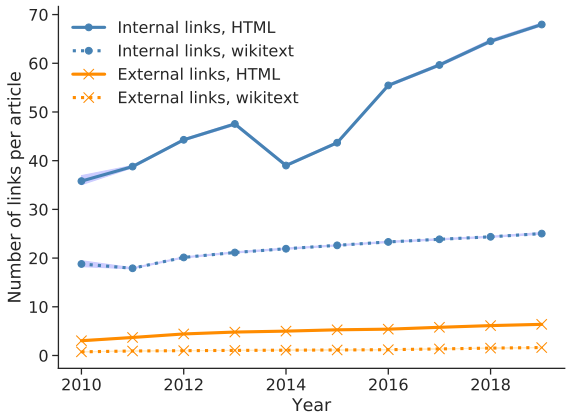

Figure 3: Number of links extracted from wikitext and HTML, averaged over 404K articles created in 2009; 95% error bands estimated via bootstrap resampling.

In this section, we illustrate the shortcomings of wikitext by showing that a large fraction of the hyperlinks apparent in the parsed HTML versions of Wikipedia articles are not visible in wikitext, thus providing researchers with a strong argument for using WikiHist.html instead of raw wikitext dumps whenever their analyses require them to account for all hyperlinks seen by readers (Dimitrov et al. 2016; 2017; Paranjape et al. 2016; West and Leskovec 2012).

Prevalence of HTML-only links over time. First we quantify the difference in the number of links that can be extracted from the wikitext vs. HTML versions of the same article revisions. To be able to determine whether the difference has increased or decreased with time, we study the 10 years between 2010 and 2019. In order to eliminate article age as a potential confound, we focus on the 404K articles created in 2009. For each article created in 2009, we study 10 revisions, viz. the revisions available at the start of each year between 2010 and 2019. For each revision, we extract and count internal links (pointing to other English Wikipedia articles) as well as external links (pointing elsewhere) in two ways: (1) based on the raw wikitext, (2) based on the HTML available in WikiHist.html.[13]

Fig. 3 shows the number of links per year averaged over the 404K articles, revealing a large gap between wikitext and HTML. The gap is significant (with non-overlapping error bands) for both internal and external links, but is much wider for internal links. Notably, for most years we can extract more than twice as many links from HTML as from raw wikitext, implying that researchers working with raw wiki-

[10] https://www.mediawiki.org/wiki/Extension:ParserFunctions

[11] https://www.mediawiki.org/wiki/Extension:Scribunto

[12] https://en.wikipedia.org/w/index.php?title=Docker_(software)&oldid=934492701

[13] As internal links, we consider only links pointing to articles in the main namespace and without prefixes, thus excluding talk pages, categories, etc. We exclude self-loops. In all analyses, if the same source links to the same target multiple times, we count the corresponding link only once. To extract internal links from wikitext, we used a regular expression crafted by Consonni, Laniado, and Montresor (2019).

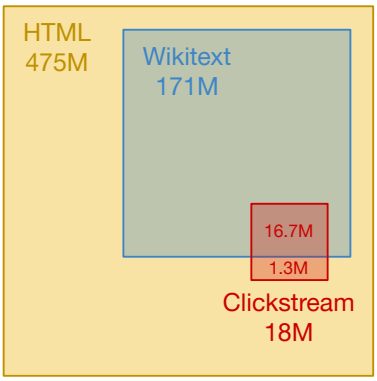

Figure 4: Venn diagram of number of links in wikitext and HTML revisions of 1 January 2019, and in Clickstream release of January 2019.

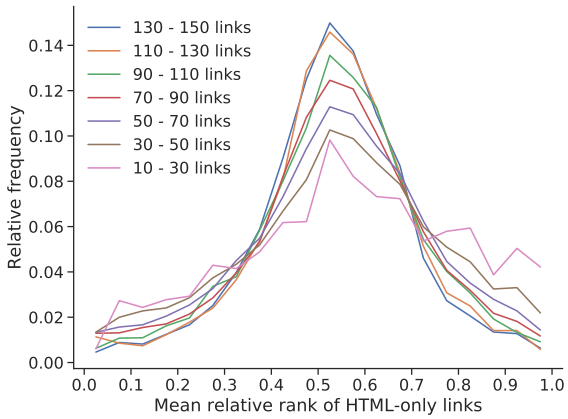

Figure 5: Histograms of mean relative rank of HTML-only links among all HTML links in terms of click frequency, averaged over 405K articles. One curve per out-degree bracket.

text (presumably the majority of researchers at present) see less than half of all Wikipedia-internal links.

Via manual inspection we found that most of the links available in HTML only (henceforth "HTML-only" links) are generated by templates and Lua modules to enhance the navigation, e.g., in infoboxes on the top right of pages or as large collections of related links at the bottom of pages.[14]

Popularity of HTML-only links. Next we aim to determine how important HTML-only links are from a navigational perspective, operationalizing the importance of a link in terms of the frequency with which it is clicked by users of Wikipedia. If, for argument's sake, HTML-only links were never clicked by users, these links would be of little practical importance, and the necessity of working with WikiHist.html rather than raw wikitext dumps would be less pronounced. If, on the contrary, HTML-only links were clicked as frequently as links also available in wikitext, then researchers would see a particularly skewed picture by not observing over half of the available links.

Click frequency information is publicly available via the Wikipedia Clickstream dataset,[15] which counts, for all pairs of articles, the number of times users reached one article from the other via a click, excluding pairs with 10 or fewer clicks. We work with the January 2019 Clickstream release.[16]

The situation is summarized as a Venn diagram in Fig. 4. On 1 January 2019, there were 475M internal links in WikiHist.html (extracted from 5.8M articles). Out of these, only 171M (36%) are also present in wikitext, and 18M (3.8%) are present in the Clickstream (i.e., were clicked over 10 times in January 2019). Strikingly, out of the 18M links present in the Clickstream, 1.3M (7.2%) cannot be found in wikitext, accounting for 6.1% of all article-to-article clicks recorded in the Clickstream. That is, joining Clickstream statistics with the contents of the respective articles is not fully feasible when working with raw wikitext. With WikiHist.html, it is.

We now move to quantifying the navigational importance of the 1.3M Clickstream links available in HTML only, relative to the set of all 18M Clickstream links available in HTML. (In this analysis, we consider only the 18M links present in the Clickstream.) For each of the 405K articles containing at least one HTML-only link, we sort all links extracted from WikiHist.html by click frequency, determine the relative ranks of all HTML-only links, and average them to obtain the mean relative rank of HTML-only links in the respective article. In the extreme, a mean relative rank of zero (one) implies that the HTML-only links are the most (least) popular out-links of the article.

Fig. 5 shows histograms of the mean relative rank of HTML-only links. To exclude the total number of out-links as a confound, we stratify articles by the number of out-links and draw a separate histogram per stratum. If HTML-only links were the least important links, the histograms would show a sharp peak at 1; if HTML-only links were no different from the other links, the histogram would show a sharp peak at 0.5. We clearly see that reality resembles the latter case much more than the former case. From a navigational perspective, HTML-only links are as important as the links also present in wikitext, and to disregard them is to neglect a significant portion of users' interests.

Beyond hyperlinks. This section illustrated the added value of WikiHist.html over raw wikitext dumps using the example of hyperlinks, but hyperlinks are not the only information to remain hidden to researchers working with wikitext only.

[14] The noticeable dip in 2014/2015 of the number of internal links extracted from HTML (top, blue curve in Fig. 3) was caused by the introduction of a then-popular Lua module called *HtmlBuilder*, which, among other things, automated the insertion of certain links during wikitext-to-HTML parsing. The module was later deleted and could not be recovered (cf. Sec. 2.4), thus leading to those links being unavailable in WikiHist.html and therefore to an underestimation of the true number of links present during the time that *HtmlBuilder* was active.

[15] https://dumps.wikimedia.org/other/clickstream/

[16] Since redirects have been resolved in the Clickstream, we also do so for links extracted from wikitext and HTML in this analysis.

Templates and modules invoked during the parsing process may also add tables, images, references, and more.

5 Conclusion

To date, Wikipedia's revision history was available only in raw wikitext format, not as the HTML that is produced from the wikitext when a page is requested by clients from the Wikipedia servers. Since, due to the expansion of templates and modules, the HTML seen by clients tends to contain more information than the raw wikitext sources, researchers working with the official wikitext dumps are studying a mere approximation of the true appearance of articles.

WikiHist.html solves this problem. We parsed English Wikipedia's entire revision history from wikitext (nearly 1 TB bzip2-compressed) to HTML (7 TB gzip-compressed) and make the resulting dataset available to the public.

In addition to the data, we also release the code of our custom architecture for parallelized wikitext-to-HTML parsing, hoping that other researchers will find it useful, e.g., for producing HTML versions of Wikipedia's revision history in languages other than English.

References

Consonni, C.; Laniado, D.; and Montresor, A. 2019. WikiLinkGraphs: A complete, longitudinal and multi-language dataset of the Wikipedia link networks. In *Proceedings of the 13th International AAAI Conference on Web and Social Media*.

Dimitrov, D.; Singer, P.; Lemmerich, F.; and Strohmaier, M. 2016. Visual positions of links and clicks on Wikipedia. In *Proceedings of the 25th International Conference on World Wide Web*.

Dimitrov, D.; Singer, P.; Lemmerich, F.; and Strohmaier, M. 2017. What makes a link successful on Wikipedia? In *Proceedings of the 26th International Conference on World Wide Web*.

Lemmerich, F.; Sáez-Trumper, D.; West, R.; and Zia, L. 2019. Why the world reads Wikipedia: Beyond English speakers. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*.

Paranjape, A.; West, R.; Zia, L.; and Leskovec, J. 2016. Improving website hyperlink structure using server logs. In *Proceedings of the 9th International ACM Conference on Web Search and Data Mining*.

Singer, P.; Lemmerich, F.; West, R.; Zia, L.; Wulczyn, E.; Strohmaier, M.; and Leskovec, J. 2017. Why we read Wikipedia. In *Proceedings of the 26th International Conference on World Wide Web*.

West, R., and Leskovec, J. 2012. Human wayfinding in information networks. In *Proceedings of the 21st International Conference on World Wide Web*.