A Dataset for Content Error Detection in Web Archives

Johannes Kiesel
Bauhaus-Universität Weimar
Weimar, Germany
johannes.kiesel@uni-weimar.de

Fabienne Hubricht
Bauhaus-Universität Weimar
Weimar, Germany
fabienne.hubricht@uni-weimar.de

Benno Stein
Bauhaus-Universität Weimar
Weimar, Germany
benno.stein@uni-weimar.de

Martin Potthast
Leipzig University
Leipzig, Germany
martin.potthast@uni-leipzig.de

1 INTRODUCTION

The World Wide Web is the single largest repository of digital culture and knowledge. Given both ubiquitous availability and the constant stream of new and noteworthy content competing for the attention of web users, one can easily miss that, at any time, reams of "old" content disappears—because web pages as well as entire websites are updated, restructured, and deleted. To preserve this content, large-scale web archiving initiatives have been started [2, 7]. The Internet Archive[1] is among the most prominent, long-lasting, and largest of such organizations; as of April 2019, its web archive collection surpassed the gigantic amount of 730 billion archived web pages.[2]

But archiving modern web pages is challenging, and a clear concept of possible errors is still missing. To further improve current web archiving technology, this paper introduces the concept of *content errors*, which refers to web pages whose archived versions have unexpected content different from their originals. This paper presents the first large scale analysis of a web crawl of 10.000 pages for content errors—the Webis Web Archive 2017 [9]. Using manual inspection and small annotation studies, we identified 5 different classes of content errors, and then annotated the entire crawl for these classes using crowdsourcing: error messages (4.5% of pages), pop-ups (3.9%), pages that largely consist of advertisements (1.1%), CAPTCHAs (0.8%), and loading indicators (0.5%). Combined, about 10% of pages are affected by content errors, which underlines the relevance of the problem. Given the large amount of web pages archived every day by the aforementioned initiatives, the detection of archiving errors in real time becomes crucial: content errors that are detected later on may not be repaired anymore, since the original page resources probably have disappeared by then.

As a step towards the automated detection of content errors at the time of archiving, we release the crowdsourced annotations as supplemental dataset to the Webis Web Archive 2017.[3] The annotations can also be visually explored using our web service at https://wwa17.webis.de. As the Webis Web Archive 2017 contains the crawled web pages as HTML DOM, screenshot, and in WARC archives, the presented annotations allow researchers to develop, test, and compare content error detection technology using features based on all information that is available to an archiving tool, even the bare HTTP messages that were exchanged during the page's archiving.

2 CONTENT ERRORS

From the perspective of the user of an archiving tool, we say that a *content error* occurred if a to-be-archived URL yields a web page that is different from what the user expected. In particular, content errors occur (1) before or during archiving, (2) are always linked to single pages, and (3) depend on what the user sees as "normal" content for the page. This definition is in contrast to, for example, spam pages [6], for which a user would not expect content in the first place. It also distinguishes content errors from reproduction errors, which occur due to incomplete archiving [3, 5, 9]. For the most part, archiving tools cannot prevent content errors, but only detect and then alert the user about them. In some cases, recovering from content errors is possible (e.g., by trying again later, investing more time, or automatically closing pop-ups). However, even the sophisticated archiving tools that are currently used by the Internet Archive or other initiatives (e.g., [1, 9, 12, 13]) perform no error detection at this moment. Nevertheless, some start to employ browser automation technology, which opens the door for automatic rectification of some types of content errors—especially pop-ups—in the future.

A classification by content error requires an error model that captures what a user does or does not expect. We adopt a *page-agnostic error model*: our hypothesized archiving tool user has a list of web page states and elements they do not expect on any archived web page (e.g., error messages). This is in contrast to a *page-specific error model*, where sets of erroneous states may be defined for an individual web page or website. For generic web archiving, devising page-specific error models may only be feasible for important pages. Our model results from a manual assessment of a sample of 300 screenshots of web pages contained in the Webis Web Archive 2017, combined with the results of two pilot crowdsourcing studies that preceded the one described in Section 3:

Error messages. The web page is not displayed correctly as indicated by an *explicit* error message. Clearly, a web page may also be displayed incorrectly without an error message, however, since a detection of this case would require prior knowledge of the "normal" state of the web page, i.e., a page-specific error model, we restrict ourselves to detect explicit error messages. A frequent cause are web pages that no longer exist, but where the server returns a substitute page with an error message (so called *soft 404* [4]). We distinguish web pages where the error message replaces the content (label: very), where the web page is still usable (label: a bit), and without error messages (label: not). Both *a bit* and *very* are content errors, as they suggest that the page's functionality is impaired.

[1] https://archive.org/

[2] https://twitter.com/brewster_kahle/status/1118172506777509890

[3] The Webis Web Archive 2017 is available at https://doi.org/10.5281/zenodo.1002204 (includes web archive files, screenshots, and DOM trees for each page), whereas the annotations gathered in this paper are available at https://doi.org/10.5281/zenodo.2549837.

CAPTCHAs. The web page asks the user to perform a task that is easy to solve for humans and supposedly very difficult for algorithms in order to block bots [14]. Archiving tools should give a warning for CAPTCHAs, so that their users can inspect the page and decide how to cope with the situation. A CAPTCHA may prevent access to the web page's content (label: very) or just prevent certain actions (most often registration and commenting) on an actually well-working page (label: a bit). Thus only *very* signals a content error.

Pop-ups. The web page shows a pop-up (e.g., overlay, banner, or modal). We distinguish pop-ups that prevent interaction with the page until closed (label: very) from those that do not (e.g., banners, cookie hints, or service chats; label: a bit). Pop-ups lead to problems as some websites load content after closing them only. Pop-ups can also derange user simulation scripts that are used in web archiving to request all relevant resources [9]. If detected, the web archiving tool may try to automatically close the pop-up. Only *very* is a content error, as for *a bit* the main functionality of the web page is still intact.

Ad page. The web page shows no real content but only ads. Such pages include domain parking pages, but also pages set up under a name similar to that of a well-known site to catch traffic arising from misspellings. There is no reason to keep them in an archive. This is a binary decision, so we distinguish *yes* and *no* only.

Loading indicators. The web page has not been fully loaded and some placeholder is shown to signal that resources are still being loaded. Note that missing resources are not archived, as well. Loading indicators can usually be resolved by prolonging the archiving (or they turn into error messages if loading fails). This kind of error is relatively rare in our dataset as the employed archiving tool uses browser automation to scroll down the web page—thereby triggering all resources to be indeed requested—and then to wait for network traffic to cease [9]. As the annotator agreement for three classes (as used for error messages, CAPTCHAs, and pop-ups) was very low for loading indicators, we distinguish *yes* and *no* only.

3 ANNOTATION PROCESS

Using the aforementioned error model, we employed crowd workers to construct the first dataset of content errors. The Webis Web Archive 2017 [9] contains 10,000 web pages sampled from the Common Crawl [11] in a way which ensured that both well-known and less-known websites are included. Table 1 shows the distribution of content errors we identified in the web pages. Every web page was annotated by at least 5 different annotators who we recruited using Amazon's Mechanical Turk. The annotation interface contained a scrollable and zoomable screenshot of the web page, radio buttons to label the content errors, and a text box for comments.

For quality assurance, we monitored annotators closely. If they took less than 10 seconds for a web page or mostly disagreed with others, we took a closer look at their annotations and—if we came to the conclusion they did not work honestly—rejected their results to be replaced by other annotators. About 10% of annotations were rejected, and a total of 747 unique annotators were recruited. Employing MACE [8], we measure a high worker agreement on all but ad pages, where only 0.65 agreement is achieved. To further improve consistency, we manually checked all cases where the annotators did not largely agree on a category and corrected the annotations, if necessary. In total, we changed 1226 annotations in this step.

Table 1: Annotator agreement [8], post-annotation corrections, distribution of labels (content errors marked bold), and percentage of pages with the respective content error identified in the 10,000 web pages of the Webis Web Archive 2017.

Content error	Agreement	Corrections	Distribution			% Error
				No	Yes	
Ad page	0.65	329		9895	**105**	1.1
Loading indicators	0.89	48		9950	**50**	0.5
			Not	A bit	Very	
Pop-ups	0.82	394	9297	315	**388**	3.9
CAPTCHAs	0.91	124	9865	60	**75**	0.8
Error messages	0.89	331	9554	**83**	**363**	4.5

4 CONCLUSION

This paper defines content errors and shows that they are not uncommon, as they appear in roughly 10% of the web pages in the broadly sampled dataset we employ. Our crowdsourced annotations for 10,000 web pages presents the first step towards an automatic detection of content errors. We envision that such automatic detectors will be used as part of web archiving tools to alert their users of the errors or even to resolve them automatically. The analysed dataset allows to devise features using the text content, HTML DOM, screenshot, or HTTP messages exchanged for a page. For error detection, where automatic approaches already exist (e.g., [10]), the presented dataset allows to incorporate new features and re-evaluate on a more recent dataset. For other types of errors, the dataset allows to develop new approaches in the first place. Therefore, the dataset allows to improve over existing methods. As content errors are noise to many analyses of web pages, detecting such errors will benefit other applications, for example web search, as well.

REFERENCES

[1] B. R. Ayala, M. E. Phillips, and L. Ko. Current Quality Assurance Practices in Web Archiving. Technical report, University of North Texas Libraries, Aug. 2014.

[2] J. Bailey, A. Grotke, E. McCain, C. Moffatt, and N. Taylor. Web Archiving in the United States: A 2016 Survey. *National Digital Stewardship Alliance*, 2017.

[3] V. Banos and Y. Manolopoulos. A Quantitative Approach to Evaluate Website Archivability Using the CLEAR+ Method. *IJDL*, 17(2):119–141, June 2016.

[4] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic Transit Gloria Telae: Towards an Understanding of the Web's Decay. In *Proceedings of WWW*, pages 328–337, 2004.

[5] J. Brunelle, M. Kelly, H. S. Eldeen, M. C. Weigle, and M. L. Nelson. Not All Mementos Are Created Equal: Measuring the Impact of Missing Resources. *IJDL*, 16(3-4):283–301, May 2015.

[6] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Inf. Retr.*, 14(5):441–465, 2011.

[7] D. Gomes, J. Miranda, and M. Costa. A Survey on Web Archiving Initiatives. In S. Gradmann, F. Borri, C. Meghini, and H. Schuldt, editors, *Proceedings of TPDL*, pages 408–420. Springer, Berlin, 2011.

[8] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. H. Hovy. Learning Whom to Trust with MACE. In *Proceedings of NAACL-HLT*, pages 1120–1130, June 2013.

[9] J. Kiesel, F. Kneist, M. Alshomary, B. Stein, M. Hagen, and M. Potthast. Reproducible Web Corpora: Interactive Archiving with Automatic Quality Assessment. *JDIQ*, 10(4):17:1–17:25, Oct. 2018.

[10] L. Meneses, R. Furuta, and F. Shipman. Identifying "Soft 404" Error Pages: Analyzing the Lexical Signatures of Documents in Distributed Collections. In *Proceedings of TPDL*, pages 197–208, 2012.

[11] The Common Crawl team. January 2017 Common Crawl Archive, 2017. http://commoncrawl.org/2017/02/january-2017-crawl-archive-now-available.

[12] The Internet Archive. Brozzler, 2017. https://github.com/internetarchive/brozzler.

[13] The Internet Archive. Umbra, 2017. https://github.com/internetarchive/umbra.

[14] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *Proceedings of EUROCRYPT*, May 2003.