CONTENT-BASED CHARACTERIZATION OF THE END OF TERM WEB ARCHIVE

Mark E. Phillips
University of North Texas
USA
mark.phillips@unt.edu
0000-0002-9679-6730

Kristy K. Phillips
University of North Texas
USA
kristy.phillips@unt.edu
0000-0002-3750-3176

Sawood Alam
Internet Archive
USA
sawood@archive.org
0000-0002-8267-3326

Abstract—Since 2008, the End of Term Web Archive has been gathering snapshots of the federal web, consisting of the publicly accessible .gov and .mil websites. In 2022, the End of Term team began to package these crawls into a public dataset which they released as part of the Amazon Open Data Partnership program. In total, over 460TB of WARC data was moved from local repositories at the Internet Archive and the University of North Texas Libraries. From the original WARC content, derivative datasets were created that address common use cases for web archives. These derivatives include WAT, WET, CDX and a format called a WARC Metadata Sidecar. This WARC Metadata Sidecar includes content-based characterizations of files held in the archive, including character set, language, file format identifier, and soft 404 detection. This paper describes the decisions made in the creation of these derivatives, the technologies used, and introduces the WARC Metadata Sidecar, which presents a useful approach for creating and storing auxiliary metadata for web archives.

Keywords – web archives, End of Term Web Archive, WARC Metadata Sidecar

I. INTRODUCTION

The End of Term (EOT) Web Archive is a collection of web crawls of all publicly available federal websites on the .gov and .mil domains collected concurrently with each presidential election since 2008. This project to document the United States federal web is the result of a collaboration between the Internet Archive, the Library of Congress, the University of North Texas, and many other organizations. The archive includes four web crawls, three of which were collected during years in which a new president was elected (2008, 2016, 2020), and one that was collected in a year in which the current president was re-elected (2012). During the years in which a new president is elected, this archive serves to document the effect of the transition on public websites. When an incumbent president is re-elected, the web crawl documents any changes made to the federal web over the four years since the previous election.

In 2022, the Internet Archive and the University of North Texas began working to create the End of Term Web Archive Dataset, a more accessible dataset of the content found in the EOT Web Archive. This dataset overcomes the logistical challenges faced by users interested in using the archive for computationally-focused research and allows open access to a large, longitudinal dataset of the federal web.

The full dataset is available with a Creative Commons CC0 1.0 Universal (CC0 1.0)[1] Public Domain Dedication and is downloadable from the End of Term Website[2]. A record for the dataset is also available in the Registry of Open Data on AWS[3].

II. RELATED WORK

The idea of a metadata sidecar file is not new. Referred to as a sidecar, buddy, or connected files, they allow for additional metadata to be stored

[1] Creative Commons CC0 1.0 Universal https://creativecommons.org/publicdomain/zero/1.0/

[2] End of Term: Data https://eotarchive.org/data/

[3] Registry of Open Data on AWS https://registry.opendata.aws/eot-web-archive/

iPRES 2023: The 19th International Conference on Digital Preservation, Champaign-Urbana, IL, US.

https://hdl.handle.net/2142/121091

alongside the primary file in situations where either the primary file does not include a method for storing arbitrary metadata, or in situations where you do not want to change the original files.

Perhaps the most common sidecar file is part of the suite of specifications that formalize file formats in Adobe's Extensible Metadata Platform (XMP) [1]. Generally, these files have the extension .xmp, and are stored in the same directory as the file that they reference. An XMP sidecar file is an XML file that stores information about the original file or change instructions from non-destructive editing tools like Adobe Bridge, Adobe Lightroom, or other tools.

Within the web archiving space, several other derivative sidecar files are commonly produced that either provide easier access to data within the original Web ARChive file format (WARC), or include a processed dataset generated from those WARC files. For example, the most common derivative file generated from the WARC records is a CDX file. A CDX file, which is a column-based text file that is used to create an index of the contents of WARC files, facilitates lookup and replay of archived web resources. Two other derivative sidecar formats common in web archiving are the Web Archive Transformation (WAT) file and the Web Archive Extracted Text (WET) file. These derivative files are often named in such a way that it is clear to users which WARC files they were derived from. For example, WAT file names typically take the base WARC name and add *.wat.gz*, the WET file names add *.wet.gz*, and the CDX file names add *.cdx.gz*. These files are typically compressed with GZip, though by different means. WAT and WET files follow the same practice as WARC records and use a record-at-time compression, while the CDX files use a full file compression. Though these filename patterns are not mandatory, they are standard practice in the web archiving community, with several software packages writing these by default (hadoop-tools, cdx-indexer, others).

It is common practice to generate derivative files for web archives, in part to improve access to the underlying data stored in the primary WARC files. This is done for several reasons, the foremost being that WARC files in web archives generally require large amounts of storage that may be beyond what a researcher interested in working with the archive might have available. To cut down on file size, derivatives that only contain a portion of the dataset are generated. For example, in a WAT file, the links, link text, and HTML metadata is the content primarily extracted. This usually results in a significant decrease in the amount of storage space required, as the WAT file only contains data extracted from certain formats like HTML, while large binary files like PDF, JPEG, or MP4 files are not included. Similarly, the WET file only contains text extracted from HTML and TXT files, so the resulting derivative file is much smaller than the original WARC file. An example of the size difference can be seen in the End of Term Dataset, where a WARC file[4] from the EOT-2020 crawl has a size of 953.7 MiB, and its corresponding WAT, WET, WARC Metadata Sidecar, and CDX files have sizes of 449.5 MiB, 82.7 MiB, 40.5 MiB, and 3.9 MiB respectively.

Over the past decade, the Archives Unleashed Project [2] has developed a toolkit and services for generating and using derivative files from web archives. This project has worked to improve the capacity for researchers to use web archives in a wide variety of research areas. The ability to work with extracted derivatives generally covers a wide range of use cases and can be a great way to encourage research interest in web archives as a data source.

The Library of Congress Web Archive and the UK Web Archive (UKWA) are among a growing group of national web archiving programs that are generating sample datasets and derivatives of their collections for use by researchers and scholars [3], [4]. In some cases, institutions are not able to directly share their web archives due to copyright or other rights restrictions. These restrictions require different approaches to data sharing. One of these approaches is to share derived metadata from the source material, which enables non-consumptive use of the underlying resources. These derivatives can also help overcome challenges researchers face in working with these web archives due to their size and scale.

Perhaps the best example of an organization that provides ample derivative formats for web archives is the Common Crawl initiative. Common Crawl operates monthly web-scale crawls of primarily text-based content like HTML, TXT, and PDF files, then

[4] EOT20-20201009-crawl800_EOT20-20201009165718-00000.warc.gz

makes these crawls publicly available. In addition to the WARC content, they generate WAT, WET, CDX, and Parquet files. Parquet files provide an index of the content in a WARC file using a column-oriented storage structure. In addition to these standard derivatives, Common Crawl provides content-based characterizations of the files they harvest at crawl time. For example, for each HTML and TXT file that they harvest, they perform content-based language identification, character set detection, and MIME type detection [5]. Because this characterization is done at crawl time, Common Crawl can store these additional metadata fields as WARC '*metadata*' records inside the primary WARC file without having to store them as a sidecar file. Once extracted, Common Crawl makes these additional metadata fields available in the CDX and Parquet indexes.

III. Overview of the EOT Web Archive Dataset

The End of Term Web Archive Dataset contains the 2008, 2012, 2016, and 2020 web crawls that make up the End of Term Web Archive. These have been collocated in the Amazon cloud as part of Amazon's Open Data Program [6]. The EOT Dataset is available using standard HTTP or an S3 client for download. The dataset is grouped so that a user can decide how much data they want to download, from the entire dataset or the data from a given election cycle, to a dataset collected by a specific crawling partner within an election cycle. The primary dataset contains ARC/WARC files, with one derivative in the format WAT, WET, CDX, and WARC Metadata Sidecar (META) created for each primary file. While the formats used in the dataset described in this paper are common in the web archiving community, it is useful to introduce the formats for those interested in the dataset that are not as familiar with the formats. Additionally, the documentation for these formats can be too dense to serve as a brief introduction to them. The sections below give a brief overview of these different files and derivatives.

A. WARC

The bulk of the dataset is housed in the WARC format. This is the standard format used in the field of web archiving to store harvested data and was designed specifically for this purpose. It contains individual WARC records that are compressed with GZip and concatenated into a single WARC file. There are different WARC record types, including '*warcinfo*,' '*response*,' '*resource*,' '*request*,' '*metadata*,' '*revisit*,' '*conversion*,' or '*continuation*'. Many tools are available for reading and writing the WARC format. The WARC format is an ISO Standard (ISO 28500:2017) and is maintained by the International Internet Preservation Consortium [7].

The WARC format was standardized in 2009 and because of this, the web crawl from 2008 contains ARC files in addition to WARC files. ARC is the predecessor to the WARC format and many web archiving institutions have chosen to maintain these original formats instead of migrating them to the WARC format. The ARC format is typically supported by tools written for the WARC format because they are so similar. The EOT Dataset maintains the original file formats and does not include any format migration from ARC to WARC in cases where ARC files were created during the initial web crawl.

B. WAT

The Web Archive Transformation (WAT) derivative is generated for each of the primary files in the dataset. These files align with the primary WARC files and provide extracted metadata and link structures from HTML content. These extractions can be used for various activities where the full text of the resource is not needed but the links from that resource and their accompanying anchor text is desired. For example, WAT files are useful for building link graphs [8]. The WAT file is generally a fraction of the size of the original WARC file. To keep alignment with the original files, the *.warc.gz* from the primary WARC is changed to *.warc.wat.gz* for the WAT file.

C. WET

The Web Extracted Text (WET) derivative is extracted text content from HTML and TXT formats in the primary WARC files. This extracted text is useful in many situations where the full structure of the HTML resource is not required. They are also streamlined for processing because they do not contain records for non-HTML and non-TXT resources. This makes the overall size of a WET file much smaller than the original WARC and generally smaller than the corresponding WAT derivative. To keep alignment with the original files, the *.warc.gz* from the primary WARC is changed to *.warc.wet.gz* for the WET file.

D. CDX

A CDX file is created that contains a row-oriented index of the WARC records inside of the WARC file. Each row contains multiple pieces of information related to the harvested content. These include: the URL, a reversed and sort friendly URL format called a Sort-friendly URI Reordering Transform (SURT), a datetime, HTTP response code, and MIME type supplied by the server the resource was harvested from, the number of bytes in the WARC record's content payload, the offset in the WARC record, the payload digest, and the WARC file path. These are generally sorted using the SURT URL key and datetime columns and then further grouped together to create indexes that can drive replay systems such as Open Wayback [9] or pywb [10]. There are several common row configurations of a CDX file, including nine-field, eleven-field, and the CDXJ configuration, which allows for more arbitrary metadata to be stored beyond the typical nine or eleven fields. The layout of a CDXJ row is the sortable URL, a timestamp, and then a single-line JSON object containing additional standard metadata fields used for replay and additional fields as needed. To keep alignment with the original files, the *.warc.gz* from the primary WARC is changed to *.cdxj.gz* for the CDX file.

E. Parquet Index

The Parquet format is a column-oriented data file format designed for efficient data storage and retrieval [11]. It is used to provide a different way of accessing the data held in the CDX derivatives. The Parquet format is used in many big-data applications and is supported by a wide range of tools. This derivative allows for arbitrary querying of the dataset using standard query formats like SQL and can be helpful for users who want to better understand what content is in the EOT Dataset using tools and query languages they are familiar with.

F. WARC Metadata Sidecar

The WARC Metadata Sidecar, referred to as the META derivative, contains content-based characterizations of the WARC records. It is described in detail in the following section.

IV. WARC METADATA SIDECAR (META)

The WARC Metadata Sidecar, referred to as the META format in the EOT Dataset, was created as a way for the team to address the problem of generating and storing additional metadata for WARC Records from the primary WARC files. As explained in the background section, the concept of a metadata sidecar file is not a new idea but an implementation of an existing concept. Other derivatives like WAT and WET essentially serve a similar function to a WARC Metadata Sidecar file, though they don't contain the same information. A WARC Metadata Sidecar file contains content-based characterizations generated using tools applied to the data, rather than a simple distillation of data from the original resource, as is found in WAT and WET files.

The metadata sidecar files contain content-based characterizations of the *response* and *resource* WARC record types. The resulting metadata fields are stored in a '*metadata*' WARC record using the warc-fields format which is a key/value format used within the WARC records themselves.

For the creation of the EOT Dataset, an open-source tool written in Python called *warc-metadata-sidecar.py* [21] was created that processes a single WARC file and generates a corresponding WARC Metadata Sidecar. The resulting filename changes the *.warc.gz* extension to *.warc.meta.gz*, which keeps the new file aligned with the original in a similar way as is done with WAT, WET, and CDX files for the dataset.

Because the project required the creation of several hundred thousand WARC Metadata Sidecar files, the team made use of mrjob [12], a Python framework for writing and running distributed computing jobs using Apache Hadoop [13]. The team used a small, 5-node Hadoop cluster housed at the UNT Libraries for the processing of all the primary WARC files.

A. Character Sets

Character set detection is implemented with the Python library *Chardet: The Universal Character Encoding Detector* [14]. This library is a continuation of the work by Mark Pilgrim and his original port of the C++ universal character encoding detector from Mozilla that he called *chardet* [15]. The output of this library is a prediction of the most likely character set of the input text and the confidence that the tool has in its prediction. These two values are stored under the key Charset-Detected in the payload of the WARC metadata record.

B. Language Identification

Language identification is accomplished using the Python bindings for the Compact Language Detector 2 (CLD2) library [16]. CLD2 can detect over 80 languages in Unicode UTF-8 text and can work with either HTML or XML. It makes use of a Naive Bayes classifier and different token algorithms. This tool was originally introduced by Google as part of the Chrome browser where it is used for language detection in that application. An updated method of language identification has been introduced called CLD3, which makes use of neural networks instead of Bayesian classifiers for language prediction. The EOT team chose to work with the CLD2 implementation because it had existing functionality for working with HTML and XML formats, while CLD3 requires conversions from those formats to UTF-8 to be done outside of the library. The output of the library is a list of the predicted languages, a score for that language, if the prediction should be considered reliable, and how much of the input text is represented by that language. The metadata sidecar takes the top three languages for a resource and stores those under the key Languages-cld2 in the payload of the WARC metadata record.

C. File Format Identification

File format identification is accomplished using the tool *Format Identification for Digital Objects* (fido) [17]. This is a tool originally developed by Adam Farquhar of the British Library and now maintained by the Open Preservation Foundation. It uses signatures from the PRONOM format registry maintained by the National Archives of the UK [18]. Fido was chosen over similar tools like DROID or Siegfried because it is written in Python and would integrate easily with the other libraries used in the *warc-metadata-sidecar.py* tool. The result of this format identification library is both a MIME Type for the format and the unique PRONOM identifier. The metadata sidecar stores the PRONOM identifier under the key Preservation-Identifier and stores the MIME Type under the key Identified-Payload-Type, with a label indicating it comes from fido.

D. MIME Type

In addition to the MIME Type that is identified using fido as described above, another MIME Type detection tool is used to provide an additional data point about the MIME type. In this case the python-magic [19] library, which is a Python interface to the libmagic file type identification library, is used. The output of this tool is often at a more general level than the output of fido. For example, python-magic might identify a file as the type *text/html*, where fido might specify the format as being *application/xhtml+xml*. Both outputs are retained for instances where either the specific or more general identification is desired. The MIME type under the key Identified-Payload-Type includes a label to indicate it comes from either fido or python-magic.

E. Soft 404 Detection

Finally, to experiment with identification of the soft 404 phenomenon, this project used the Python tool Soft-404 [20]. The soft 404 phenomenon occurs when a web server responds with an HTTP response code of *200 OK*, but returns a page that indicates that the content is not available instead of returning a *404 Not Found* response code. The Soft-404 library uses a classifier that was trained on 198,801 pages from 35,995 domains, with a 404 page ratio of about 1/3. The EOT Dataset used the provided model for soft 404 detection. The result is a value between 0 and 1 that shows how likely the page is a soft 404 example with scores closer to zero being unlikely and those closer to 1 more likely to be a soft 404. This value is stored under a key of Soft-404-Detected in the payload of the WARC metadata record.

V. Discussion

There are several reasons it is a good idea to do content-based identification. One example is MIME Type identification using actual content over provided values. In this situation, a server can provide a MIME type like application/pdf and a URL such as https://example.com/sample.pdf, but because of an error in confirmation in the web server, or an error in the coding to dynamically generate content, an HTML file reporting the error or unavailability of the page (404) might be returned without reporting the correct MIME Type. Content-based identification in this case can accurately identify the actual MIME type of the content as HTML. This identification is also important for recognizing Soft 404 which can often return a 200 response with a given MIME type but, the content is text/html.

The metadata extracted from WARC records and included in the WARC Metadata Sidecar files can be used to build indexes of the content available in the End of Term Web Archive. As an example of this, we are using the content-based characterization data in

combination with the standard data found in a CDX index to build a Parquet index for each of the End of Term crawls. These Parquet indexes allow users to answer questions related to the web archives that previously would have been challenging to ask, such as ”what MIME types are misreported the most?”, ”what domains have the most misreported content?”, ”what is the prevalence of non-English content in the archive?”, ”what domains have the most non-English content?”, ”which non-English languages are most represented in the archive?”, ”how prevalent are Soft 404’s and which domains have the most instances of them?”, and ”what are the file types present based on file identifiers?” This list can easily go on, and these are questions that can be answered by writing SQL queries to interact with the Parquet index and do not require traversing the dataset as it would have in the past.

VI. Future Work

The WARC Metadata Sidecar file introduces a method for storing metadata from different content-based characterization tools and associating that metadata with the original WARC files that make up a web archive. They provide a logical alignment with WARC records and allow for content-based characterization of content in ways that were previously unavailable, or with new approaches or tools. The new metadata that is generated can be incorporated into indexes that provide opportunities to answer research questions related to large-scale web archives that could otherwise be challenging to answer.

The implementation of WARC Metadata Sidecar files in this project might be improved in several ways. First, the *warc-metadata-sidecar* tool [21], written in Python and then integrated as mrjob jobs on a Hadoop cluster, was successful at processing content at scale. Inefficiencies were recognized as more content was processed, though there are still situations where the tools might need further optimization to deal with the number of files that require characterization. Limitations exist thanks to file formats that are not compatible with the tools used for content-based language identification, like PDF and JPEG files. One way to improve the implementation of the warc-metadata-sidecar.py in the future might be to incorporate tools like the Tika library [22] to convert additional formats like PDF or Microsoft Word Documents into text that can be further characterized. The introduction of a Java-based tool to the process might warrant a change in underlying programming language used for the overall script. Another option is to investigate Python-based converters that can extract text from various binary files so that they can be incorporated into the output.

Future work for this dataset includes generation of host-level and domain-level network graphs that will show the relationships between domains within the EOT Web Archive. This work is expected to continue to leverage existing tools and processes developed by Common Crawl for graph generation. With the complete dataset available in CDX format, overviews of each of the EOT crawl years using CDX summarization tools [23] can be generated. These can be helpful in communicating the content of this dataset to others.

VII. Acknowledgements

This effort would not have been possible without storage support from the Amazon Open Data Program, which provided S3 storage for this initiative. Likewise, this project leaned heavily upon the prior work of the Common Crawl team and adopted their organizational structures, tools, and documentation in building this dataset and providing access to it.

1. REFERENCES

[1] Adobe, “Adobe – XMP developer center.” https://www.adobe.com/devnet/xmp.html, 2023.
[2] Archives Unleashed, “The Archives Unleashed Project.” https://archivesunleashed.org/, 2022.
[3] Library of Congress, “Web archive datasets,” 2023.
[4] UK Web Archive, “UK web archive open data,” 2023.
[5] Common Crawl, “August crawl archive introduces language annotations.” https://commoncrawl.org/2018/08/august-2018-crawl-archive-now-available/, 2018.
[6] Amazon.com, Inc., “Open Data Sponsorship Program,” 2022.
[7] International Internet Preservation Consortium, “The WARC format 1.0.” https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.0/, 2022.
[8] Common Crawl, “Common Crawl’s first in-house web graph,” 2017.
[9] International Internet Preservation Consortium, “OpenWayback.” https://github.com/iipc/openwayback, 2021.
[10] Webrecorder, “pywb.” https://github.com/webrecorder/pywb, 2023.
[11] The Apache Software Foundation, “Apache Parquet,” 2023.
[12] Yelp,“mrjob: the Python mapreduce library.” https://github.com/Yelp/mrjob, 2020.
[13] The Apache Software Foundation, “Apache Hadoop,” 2023.

[14] D. Blanchard, “Chardet: The universal character encoding detector.” https://github.com/chardet/chardet, 2022.
[15] M. Pilgrim, Case Study: Porting chardet to Python 3, pp. 253–277. Berkeley, CA: Apress, 2009.
[16] R. Alrfou, “PYCLD2 – Python bindings to CLD2.” https://github.com/aboSamoor/pycld2, 2022.
[17] Open Preservation Foundation, “Format identification for digital objects (fido).” https://github.com/openpreserve/fido, 2022.
[18] The National Archives, “Pronom.” https://www.nationalarchives.gov.uk/PRONOM 2006.
[19] Hupp, “python-magic.” https://github.com/ahupp/python-magic, 2022.
[20] TeamHG-Memex, “soft404: a classifier for detecting soft 404 pages.” https://github.com/TeamHG-Memex/soft404, 2017.
[21] University of North Texas Libraries, “WARC Metadata Sidecar.” https://github.com/unt-libraries/warc-metadata-sidecar, 2023.
[22] The Apache Software Foundation, “Apache Tika.” https://tika.apache.org/, 2023.
[23] S. Alam and M. Graham, “CDX summary: Web archival collection insights,” vol. 13541 of Lecture Notes in Computer Science, pp. 297–305, Springer, 2022.