“The Library is Open!”: Open Data and an Open API for the HathiTrust Digital Library

John A. Walsh[1,2,*], Glen Layne-Worthey[1,3], Jacob Jett[3], Boris Capitanu[1,3], Peter Organisciak[4], Ryan Dubnicek[1,3] and J. Stephen Downie[1,3]

[1]*HathiTrust Research Center, Indiana University Bloomington and University of Illinois Urbana-Champaign, USA*

[2]*Luddy School of Informatics, Computing, and Engineering, Indiana University, 700 N Woodlawn Ave., Rm. 2132, Bloomington IN 47408, USA*

[3]*School of Information Sciences, University of Illinois, Urbana-Champaign, 614 E. Daniel St., Champaign, IL 61820, USA*

[4]*Department of Research Methods & Information Science, University of Denver, USA*

Abstract

This paper describes the history, policy, semantics, and uses of the HathiTrust Research Center Extracted Features dataset, an open-access representation of the 17+ million volume HathiTrust Digital Library, including a major current effort to extend computational access in a variety of more flexible and easily implemented ways, including a modern API supporting customizable visualizations and analyses.

Keywords

digital libraries, cultural analytics, text analysis, APIs

1. Introduction & Context

HathiTrust,[1] founded in 2008, is a not-for-profit collaborative initiative of academic and research libraries that seeks to preserve and make accessible for reading and computation the combined corpora of digitized objects from the HathiTrust consortium’s membership. As of this writing, that corpus, known as the HathiTrust Digital Library, has now grown to contain more than 6.1 billion digitized pages of content comprising almost 17.7 million volumes of text representing nearly 9 million unique works (including more than 8.4 million books and 470 thousand serials titles).

The work described below addressed two core research problems. First, the Extracted Features (EF) dataset provides access to the HathiTrust corpus in a non-consumptive manner

CHR 2023: Computational Humanities Research Conference, December 6 – 8, 2023, Paris, France

*Corresponding author.

†These authors contributed equally.

jawalsh@indiana.edu (J. A. Walsh); gworthey@illinois.edu (G. Layne-Worthey); jacob@floods.org (J. Jett); Peter.Organisciak@du.edu (P. Organisciak); rdubnic2@illinois.edu (R. Dubnicek); jdownie@illinois.edu (J. S. Downie)

https://johnwalsh.name/ (J. A. Walsh)

0000-0002-1824-210X (J. A. Walsh); 0000-0003-2785-0040 (G. Layne-Worthey); 0000-0003-1939-6255 (J. Jett); 0000-0002-9058-2280 (P. Organisciak); 0000-0001-7153-7030 (R. Dubnicek); 0000-0001-9784-5090 (J. S. Downie)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://www.hathitrust.org/

CEUR-WS.org/Vol-3558/paper7875.pdf

that allows computational analysis of a representation of the entire corpus while respecting copyright and other restrictions on that content. The TORCHLITE project addresses a second research problem by providing efficient, programmatic access to the Extracted Features dataset. Previously, general users were required to use the rsync utility to access the EF dataset. The TORCHLITE project provides more flexible, efficient, and programmatic access to the EF dataset through modern web-based APIs. The EF dataset and the TORCHLITE APIs are open and available to all users regardless of membership or affiliation with the HathiTrust Digital Library.

One of the core missions of HathiTrust is to provide access to its collections. Towards this end, in 2011, HathiTrust established the HathiTrust Research Center[2] (HTRC), with one of its primary objectives to solve the problem of computational access to the massive HathiTrust corpus. One key service through which HTRC supports this access is the periodic publication of *Extracted Features* (EF) datasets, comprised of quantified features derived from the full-text corpus using computational analysis. The EF datasets serve researchers in a range of fields (including the humanities, social sciences, data science, and machine learning) with a foundational, structured dataset that may be used to develop analytics algorithms for computational exploration of the HathiTrust corpus. The EF datasets may be downloaded or accessed through our new TORCHLITE EF API, without restriction, as easily parsed files encoded as "JSON for Linked Data" (JSON-LD). Through these EF datasets, researchers have access to a snapshot of the entire HathiTrust corpus, in a format that provides useful, inferential fingerprints of a set of texts without giving away the original copyrighted work, a mode of access known as "non-consumptive research." Non-consumptive research is computational analysis performed on text without substantial portions of the text displayed or read to understand its expressive content.[3]

2. A Brief History of the Extracted Features Datasets

2.1. Extracted Features 0.2

The initial release[4] of EF, versioned as 0.2 [1] [4], provided researchers with an opportunity to download the unigram tokens parsed from the 4.8 million volumes (over 1.8 billion pages of text) corresponding to the portion of the HathiTrust corpus that was in the public domain for all jurisdictions in 2015. These EF files also contained basic analytical information about the text of each volume, such as sentence, line, and empty-line counts at the volume, page, and page-part (i.e., header, body, footer) levels for each volume in the dataset. This data was intended to allow the bootstrapping of additional analyses by reducing the number of steps that individual researchers needed to take to develop tools that interact with the data. One of the first examples of such bootstrapped work was the diachronic word frequency visualizations offered by the Bookworm[5] tool, which offered a simple public interface to allow users to interact with that

[2] https://www.hathitrust.org/htrc
[3] https://www.hathitrust.org/htrc_ncup
[4] https://wiki.htrc.illinois.edu/pages/viewpage.action?pageId=37322766
[5] https://bookworm.htrc.illinois.edu/

first snapshot of the corpus, and which has since been updated to include later versions of the EF dataset, about which see more below.

The unigram tokens in this dataset were parsed from HathiTrust's OCR text files using the OpenNLP[6] library. In addition to parsing the tokens, the OpenNLP library was used to characterize each token's part of speech (POS). The primary language used on each page analyzed was also calculated using Shuyo Nakatani's Language Detection library.[7] A highly structured JSON schema was developed that allowed for the presentation of all the tokens, their parts of speech, and the number of their occurrences, separated into the header, body, and footer sections of each page. Each JSON file encoded the features extracted from exactly one volume in the HathiTrust corpus.

Importantly, an unusual implementation feature of the common software libraries used for parsing JSON documents was exploited to provide a more efficient means of processing the EF data encapsulated within the individual EF files. In particular, rather than follow the typical `"key":"value"` pairing defined by the JSON standard, fields containing the tokens, POS tags, and their respective counts (i.e., the actual textual-statistical data) take advantage of the fact that JSON parsing libraries have no semantic sense of the concept of a `"key"`. Rather than treating each `"key"` as a label, these libraries parse it as an arbitrary string.[8] This helpful semantic ignorance provides an important simplifying pattern for using JSON to communicate data, rather than following a verbose pattern such as `"key":{"key":"value","key":"value","key":"value"}` — that rigorously conforms to the JSON specification, EF files employ an "off-formulary" approach using a much less verbose pattern of `"value":{"value":"value"}`. Thus, the all-important statistical part of the dataset follows the simplified pattern `"token":{"POS":"count"}`.

This non-standard approach for the `features` section of each EF file reduces both file size (by approximately half) and data complexity ("flattening" the file structure and bringing the all-important linguistic and statistical data one level "higher" in the index and parsing trees); it also makes possible the exploitation of efficiencies already built into existing JSON parsing libraries.[9] While these gains may seem trivial in most common contexts (involving hundreds or thousands of files), the current EF release weighs in at 17.1 million files and 4.2 TB (compressed), so our estimated 50% gains in storage, data transfer, and processing seem to us more than substantial.

This approach, however, requires data consumers to familiarize themselves with the EF schema and its non-standard usage to be aware of fields for which standard key-value pairs are intended, and those for which value-value pairs are used. As long as these differences in semantics are taken into account (and they should be fairly obvious and visible to users), the same JSON parsing and processing libraries can be for both data patterns; users simply need

[6] http://opennlp.apache.org

[7] https://code.google.com/p/language-detection/

[8] Note that the rest of the JSON file — that is, everything except word and POS counts — uses `key:value` pairs in a more customary fashion to denote header, metadata, and statistical information about the dataset.

[9] These estimates are based on a simple comparison of a single random value expressed in the two encodings, requiring about 50 characters in the standard encoding, and about 25 using our approach. Even in this more streamlined encoding, the `features` section of a single EF file, representing a randomly selected 220-page volume, is about 850,000 characters, as compared with the `metadata` section of only about 2,000 characters, even in the more verbose standard syntax.

to treat the outputs differently. As we discuss in the section describing the EF 2.0 dataset (below), these semantic differences also have consequences for the dataset's compliance with the Linked Data standard.

In addition to the token data for a particular volume, each EF file also included a basic set of bibliographic metadata for that volume (using the `key-value` pair pattern) describing the following information:

- volume title
- primary author
- volume imprint information (a single, unparsed field that included publisher, place of publication, year of publication)
- the OCLC[10] identifier associated with the volume's catalog record
- the primary language of the volume (as identified by human catalogers)
- the URL of the volume's HathiTrust catalog record
- the URL of the digitized volume's persistent HathiTrust identifier

With each release of the EF dataset, we have considered separating the JSON standard-compliant `metadata` section from the non-compliant but still fully functional `features` section; this consideration was especially intense with the current release (described below) in which we first created Linked Data (expressing volume and file metadata as fully compliant JSON-LD). So far, we have made the decision not to do this in favor of offering a "one-stop-shopping" single file per volume, which contains all of the data and metadata that a researcher needs to carry out analytic tasks. Maintaining our original approach also facilitates some backward compatibility with both previous EF versions and existing tools built specifically for this data.

2.2. Extracted Features 1.0 and 1.5

The subsequent EF releases (designated as versions 1.0 and 1.5)[11] [2] represented a greatly expanded corpus, including for the first time the entirety of the HathiTrust corpus at its time of creation (late 2016) — not only public domain, but also volumes that may have been subject to any of several copyright regimes, licenses, or other access restrictions. Open access to data derived from these (potentially) in-copyright volumes relies on the aforementioned "non-consumptive research" paradigm, which prevents release of any human-readable full text while still providing for a wide range of computational analyses. This dataset included about 2.5 trillion unigram tokens (and their parts of speech and occurrence counts) parsed from the almost 5.8 billion pages contained in just over 15.7 million volumes, resulting in the same number of EF JSON files.

In addition to vastly increasing the size and coverage of the dataset, the metadata section of each EF file was also significantly expanded with more complete and granular information from the MARC records describing its volumes. In addition to the basic catalog fields listed above, volume metadata also now included:

[10] OCLC, https://www.oclc.org/, formerly the Online Computer Library Center, is perhaps best known as the compiler and maintainer of WorldCat, https://www.worldcat.org/.

[11] https://wiki.htrc.illinois.edu/pages/viewpage.action?pageId=37322778

- publication place, parsed out from the imprint statement
- publisher, parsed out from the imprint statement
- publication (copyright) year, parsed out from the imprint statement
- cataloger-determined genre
- cataloger-determined resource type
- additional creator names (e.g., additional authors, editors, illustrators, etc.)
- copyright status as determined by HathiTrust (i.e., whether the volume is in the public domain in particular jurisdictions)
- additional bibliographic identifiers such as ISBN, ISSN, LCCN (Library of Congress Catalog Number), and call numbers assigned by the volume's owning institution(s)

2.3. Extracted Features 2.0

One of the original goals of creating a new EF 2.0 schema and dataset[12] [3] was to allow the use of the EF schema beyond the HathiTrust sphere: that is, to work toward a standardized data schema for extracted features that could be employed by a large number of digital content providers (e.g., publishers and aggregators). As often happens in collaborative projects, however, the expansion of stakeholders resulted in a corresponding expansion of the numbers and kinds of use cases, and in the end the collaborators were not successful in creating a shared schema. But all the partners did share a desire to employ contemporary Linked Data vocabularies in order to facilitate a richer set of metadata and data interactions with and within the Extracted Features dataset.

In addition to this move to Linked Data, EF 2.0 represents the latest set of bootstrapping features computationally mined from the HathiTrust corpus. Since the release of EF 1.0/1.5, the dataset has grown in size to just over 2.9 trillion unigrams parsed from over 6.2 billion pages contained in just over 17.1 million volumes. Like the leap from EF 0.2 to EF 1.0/1.5, the new EF 2.0 dataset further expands, while also refining, the volume-level metadata provided within each EF file. In particular, the initial development of the JSON schema underlying the EF 2.0 model was undertaken in collaboration with the JSTOR and Portico projects[13] and while not all of the innovations developed during this collaboration were implemented into EF 2.0 in the end, the switch from MARC metadata as the basis for volume-level metadata in EF files to the more refined, linked-data-compliant BIBFRAME[14] records made it possible, for instance, to link EF files describing different instances of the same work to one another.

In the following sections we delve more deeply into how the EF 2.0 dataset differs from older EF datasets, and how it continues in their tradition.

2.3.1. EF 2.0 Metadata

The primary innovations for the metadata section of EF 2.0 files involve the implementation of linked data practices. First and foremost, a subtle change in the file format from JSON to

[12] EF 2.0 is the current version as of the time of writing: https://wiki.htrc.illinois.edu/pages/viewpage.action?pageId=79069329

[13] Both JSTOR, https://www.jstor.org/, and Portico, https://www.portico.org/, are projects under the nonprofit ITHAKA umbrella, https://www.ithaka.org/.

[14] https://www.loc.gov/bibframe/.

JSON-LD was made. Unlike ordinary JSON files which use arbitrary vocabulary structures to manage their key labels, JSON-LD employs a type of schema document known as a *context document* to establish the semantics for each JSON-LD file. Each context file works like an XML schema, defining the semantics for each key label or linking its semantics to an existing ontology or metadata vocabulary standard. To fill this requirement, HTRC developed a robust context document[15] with two primary purposes: first, to linked the semantics of various EF key labels to terms in the Schema.org vocabulary;[16] and second, to formally name extensions to the Schema.org vocabulary that were created explicitly for use with EF data files.

This change from "ordinary" metadata to linked data afforded the opportunity for the HTRC to exploit previous work transforming the HathiTrust catalog of MARC metadata records into linked data records conforming to the Library of Congress's (LoC) BIBFRAME metadata ontology [3].[17] While the originating source of the volume-level metadata of each EF 2.0 file remains the same as it was for previous releases (viz., the MARC records of contributing libraries), the transformation into BIBFRAME allowed us to apply additional metadata quality control work by reconciling existing contributor and certain other fields with external authorities databases such as the Virtual International Authority File (VIAF), which make widely available linked data URIs identifying named entities such as people, places, and organizations. A workflow for reconciling works and instances using OCLC work identifiers was also implemented to better link instances of the same work to one another. Jett, et al., (2020b) describe both the MARC-to-BIBFRAME conversion process and the reconciliation work in greater detail.

In addition to the named entity reconciliation work undertaken to create richer linked data, the EF 2.0 metadata section was further expanded to incorporate additional metadata to support research in the humanities and a variety of other fields. These newly added metadata details include:

- Additional links to brief and long metadata records in the HathiTrust catalog.
- Finer-grained information regarding journals and monographic series, including which issue and volume numbers make up a particular digitized volume).[18]
- Plain-text classification descriptions for Library of Congress call numbers, when they are included in the original records, mapped directly from the LoC's documentation.[19]

2.3.2. EF 2.0 Data

The JSON-LD approach works wonderfully in the metadata section of EF files by enabling a variety of linked data affordances, and by forcing a distinction between the metadata describing the EF files themselves, the metadata describing the volume from which the extracted features were derived, and the metadata describing the features themselves (e.g., token and part-of-speech counts. However, the stricter rules of JSON-LD work less well for the data section of the document. Fortunately, the realities of processing JSON and JSON-LD documents allow any

[15] https://worksets.htrc.illinois.edu/context/ef_context.jsonld
[16] https://schema.org/
[17] https://www.loc.gov/bibframe/
[18] The semantics here are subtly different from "imprint," which contains this information in human-readable format. The finer grained "issue" and "volume" keys make this data machine-readable.
[19] https://id.loc.gov/authorities/classification.html

software capable of parsing a JSON document to be used to parse the EF JSON-LD document as well, by simply ignoring the ”LD” semantics that might be used by Semantic Web tools. Applying an EF 2.0 file's context document to the whole of the file will end up corrupting the data due to the efficiency-making measures designed into the 0.2 and 1.0/1.5 versions of the dataset's schemas. However, it is not the case that the context document need be applied to the whole of each EF 2.0 file (or to any of it at all).

Context documents exist to enforce document semantics by mapping the keys to predicates in specified ontologies, like Schema.org or BIBFRAME, and the various data values to objects and subjects that are focal points and data values within the scope of those ontologies. However, in most cases it will not be necessary to apply the context document to the EF 2.0 file itself. If one chooses to apply a context document, for instance, because one wants to add volume-level descriptive metadata to a triple store, this can be simply done by separating the metadata section from the data section and then applying the context document to just the metadata section of the file to produce linked data triples in JSON format, which most triple store software can consume directly. Applying the context document to the data section of the EF file will cause corruption of the data within the section because applying the context document will attempt to transform all of the ”key” values into URLS, even where the pattern used for the data is ”value-value” instead of ”key-value.”

With the exception of the above caveats, no significant changes were made to the data part of the EF schema (though some typographical errors in key labels that had persisted since the 0.2 version were corrected). Instead, the primary change to the data section in EF 2.0 concerns how the HTRC generates the extracted features data. Instead of employing the OpenNLP library as its primary means for parsing the OCR text files, we switched to the Stanford NLP library[20] to parse and count the unigram tokens, and to apply POS tags to them. This change in NLP libraries necessitated a change in the tagsets being used for part-of-speech tags — from the OpenNLP POS tagset to the Stanford NLP tagset. The rest of the data supplied as part of EF datasets, including line counts, empty line counts, sentence counts, tokens, token POS tags, and associated counts, all remained consistent with the EF 2.0 release.

3. Leveraging intelligent text extraction: TORCHLITE

We conclude with a description of a major new HTRC effort intended to make its rich EF data more accessible, more usable, and more readily available to researchers and librarians: *Tools for Open Research and Computation with HathiTrust: Leveraging Intelligent Text Extraction* (TORCHLITE), generously funded by the U.S. National Endowment for the Humanities in 2021.[21]

The TORCHLITE project is intended to enhance HTRC's current data delivery infrastructure with a new, non-relational (“NoSQL”) database customized to host the EF dataset and a robust and well-documented API for accessing the data within it. The API allows for retrieving highly targeted subsets of the EF dataset, down to the level of individual volumes or pages, if desired. For example, using a HathiTrust volume ID, one can retrieve the complete EF data for the

[20] https://stanfordnlp.github.io/stanfordnlp/models.html
[21] Grant no. XXX

corresponding volume, and with a HathiTrust volume ID and a sequential page number, one could retrieve the EF data for a specific page. These subsets of the EF dataset can be gathered into user-defined "worksets," which are simple lists of HathiTrust volume IDs (optionally with specific page numbers).

Not only does the API enable the retrieval of all the EF data for a particular workset of volumes (or pages); it also enables retrieval of specific, granular subsets of the EF data for that set of volumes, including volume-level metadata elements (e.g., title, publisher, publication date, genre, page count, etc.); page-level metadata such as algorithmically-determined page language; and page-level token counts, parts of speech, and line and sentence counts.

This API is intended to feed lightweight analytical tools ("widgets") that can be incorporated into any modern website — for example, that of a library catalog or digital scholarly publication. These widgets can also be combined into a "dashboard," together data selection and cleaning tools, statistical information about the workset that is represented in the various widgets (e.g., total word count, most frequent words, vocabulary density), word frequency visualizations (e.g., word clouds, charts comparing frequencies of user-selected words across volumes), and comparison of multiple volumes in that workset. The dashboard and widgets are built with modern, open data visualization libraries (e.g., chart.js, D3.js, etc.). A mockup of the TORCHLITE dashboard with two sample widgets is shown in Figure 1.

The TORCHLITE dashboard will be part of the HTRC Analytics site at https://analytics.hathitrust.org/, accounts for which are freely available to anyone from an educational or research institution, regardless of HathiTrust membership status, but the API is also open to others to access EF data for use in their research and tool development. The data and metadata selection and cleaning parameters, together with whatever statistical protocols are employed in a particular widget, can also be downloaded as an independent Jupyter notebook for further customization and manipulation by researchers.

3.1. Improving access to extracted features

Before the TORCHLITE project, EF data was accessed either through the standard `rsync` file transfer tool[22] or through a custom Python library called the HTRC Feature Reader.[23] While these modes of access are serviceable and support individual researchers who want to download and process the data, they require of those researchers a certain amount of programming skills, and do not integrate well with the modern Web environment or current development trends, which assume simple API-based data access.

As an initial step towards building a data dashboard that would allow users to analyze and visualize subsets of the HathiTrust collection, HTRC created a MongoDB[24] to store a copy of the EF data and developed a simple API for accessing the data.

[22] https://wiki.htrc.illinois.edu/display/COM/Downloading+Extracted+Features
[23] https://github.com/htrc/htrc-feature-reader
[24] https://www.mongodb.com

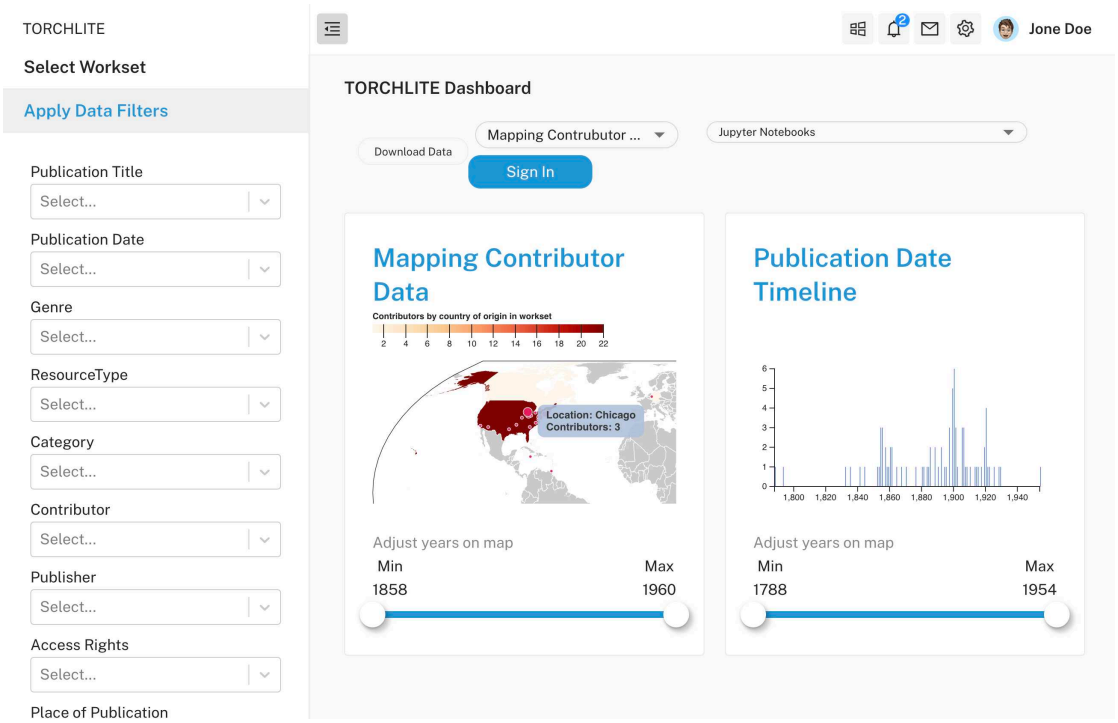

Figure 1: Mockup of the TORCHLITE dashboard with two widgets: on the left, one that combines author data and publication date from the HathiTrust Extracted Features with biographical and geographical data from Wikidata, producing an interactive map of the birthplaces of authors included in a small sample workset of American fiction; on the right, one that displays a histogram of publication dates represented in that workset.

3.2. TORCHLITE API

The TORCHLITE API is a RESTful API with a few simple calls for creating and retrieving worksets and retrieving EF data. For instance, the following call will retrieve the volume metadata for a single volume: `https://tools.htrc.illinois.edu/ef-api/volumes/{clean-htid}/metadata`. By replacing `{clean-htid}` with the HathiTrust volume ID for a volume of poetry by Victorian poet Algernon Charles Swinburne, we can view the volume metadata for a copy of Swinburne's *Poems and Ballads* (1866): https://tools.htrc.illinois.edu/ef-api/volumes/uc2.ark+=13960=t17m0815m/metadata. Below is the metadata retrieved in response to the above call:


```
{
  "code": 200,
  "data": {
    "htid": "uc2.ark:/13960/t17m0815m",
    "metadata": {
      "schemaVersion": "https://schemas.hathitrust.org/EF_Schema_MetadataSubSchema_v_3
    .0",
      "id": "http://hdl.handle.net/2027/uc2.ark:/13960/t17m0815m",
```

```
      "type": [
        "DataFeedItem",
        "Book"
      ],
      "dateCreated": 20200209,
      "title": "Poems and ballads. /",
      "contributor": {
        "id": "http://www.viaf.org/viaf/41846937",
        "type": "http://id.loc.gov/ontologies/bibframe/Person",
        "name": "Swinburne, Algernon Charles, 1837-1909."
      },
      "pubDate": 1866,
      "publisher": {
        "id": "http://catalogdata.library.illinois.edu/lod/entities/
    ProvisionActivityAgent/ht/John%20Camden%20Hotten,%20Piccadilly",
        "type": "http://id.loc.gov/ontologies/bibframe/Organization",
        "name": "John Camden Hotten, Piccadilly"
      },
      "pubPlace": {
        "id": "http://id.loc.gov/vocabulary/countries/enk",
        "type": "http://id.loc.gov/ontologies/bibframe/Place",
        "name": "England"
      },
      "language": "eng",
      "accessRights": "pd",
      "accessProfile": "open",
      "sourceInstitution": {
        "type": "http://id.loc.gov/ontologies/bibframe/Organization",
        "name": "INRLF"
      },
      "mainEntityOfPage": [
        "https://catalog.hathitrust.org/Record/100629227",
        "http://catalog.hathitrust.org/api/volumes/brief/oclc/16440752.json",
        "http://catalog.hathitrust.org/api/volumes/full/oclc/16440752.json"
      ],
      "oclc": "16440752",
      "genre": "http://id.loc.gov/vocabulary/marcgt/doc",
      "typeOfResource": "http://id.loc.gov/ontologies/bibframe/Text",
      "lastRightsUpdateDate": 20181212
    }
  }
}
```


A subset of fields may be retrieved with the `fields` query parameter, e.g.: https://tools.htrc.illinois.edu/ef-api/volumes/uc2.ark+=13960=t17m0815m/metadata?fields=metadata.pubDate,metadata.pubPlace. This API call retrieves the following data:


```
{
  "code": 200,
  "data": {
    "metadata": {
      "pubDate": 1866,
      "pubPlace": {
        "id": "http://id.loc.gov/vocabulary/countries/enk",
```

```
        "type": "http://id.loc.gov/ontologies/bibframe/Place",
        "name": "England"
      }
    }
  }
}
```

This API call retrieves all of the EF data for a single volume: `https://tools.htrc.illinois.edu/ef-api/volumes/{clean-htid}`

As in the previous example, replacing `{clean-htid}` with the HathiTrust volume ID for Swinburne's *Poems and Ballads*, we can retrieve the EF data for the entire volume of poetry: https://tools.htrc.illinois.edu/ef-api/volumes/uc2.ark+=13960=t17m0815m. The output includes tokens and token counts for every page in the volume, and is thus too long to include here, but the data may be viewed by following the above link. As noted above, a subset of metadata and data fields may be retrieved using the `fields` query parameter.

Additional calls are documented in the complete API documentation at https://htrc.stoplight.io/docs/torchlite/. While not yet widely promoted, the TORCHLITE API is now open and available to users and developers. In May 2024, we will hold a developer-focused "Hackathon" event to encourage the adoption of the API by digital humanities researchers and developers and others applying text-mining methods to their research. Also in 2024, we will release the public version of the Dashboard, which is currently still in development and testing.

3.3. Including external data sources

One particularly exciting feature of TORCHLITE's API-based architecture is the new ability to combine EF data with non-HTRC data, retrieved through other APIs, into richer visualizations and other types of analysis. The widget shown on the left side of Figure 1, for example, combines publication dates and author names from a workset of Extracted Features (all retrieved via the TORCHLITE API), with geographic and biographic data about those authors (retrieved from Wikidata via its API), to map the birthplaces of authors represented in the workset.

4. Future Work

As with past versions of the Extracted Features dataset, future iterations will continue to include new volumes as they are continually added to the HathiTrust corpus and incorporate new statistical measures, data features, and functionality based on the state of the art in cultural analytics and linked data research.

For the more immediate future, the TORCHLITE project is refining its existing suite of widgets and expanding it to include new types of analysis and visualization. Since an important goal of the project is to allow for community-conceived, community-created, and even externally-hosted widgets, the project also has plans to host a workshop and hackathon to attract scholars and developers from the HathiTrust user community interested in experimenting with the vastly improved access to the rich open data that HTRC has made available from the HathiTrust Digital Library.

References

[1] B. Capitanu, T. Underwood, P. Organisciak, S. Bhattacharyya, L. Auvil, C. Fallaw, and J. S. Downie. *Extracted Feature Dataset from 4.8 Million HathiTrust Digital Library Public Domain Volumes (0.2)*. WebPage. 2015. DOI: 10.13012/j8td9v7m.

[2] B. Capitanu, T. Underwood, P. Organisciak, T. Cole, M. J. Sarol, and J. S. Downie. *The HathiTrust Research Center Extracted Feature Dataset (1.0)*. WebPage. 2016. DOI: 10.13012/j8x63jt3.

[3] J. Jett, D. Kudeki, G. Worthey, T. Cole, and J. S. Downie. "Applying BIBFRAME in large-scale digital libraries: The HathiTrust Research Center's experience". In: *Proceeding of the 83rd ASIS&T Annual Meeting*. virtual: Asis&t, 2020.

[4] P. Organisciak, B. Capitanu, T. Underwood, and J. S. Downie. "Access to Billions of Pages for Large-Scale Text Analysis". In: *Proceeding of the 2017 iConference*. Wuhan, China: iSchools, 2017. URL: https://hdl.handle.net/2142/98491.