EXPLORING THE BILLIONS AND BILLIONS OF WORDS IN THE HATHITRUST CORPUS WITH BOOKWORM: HATHITRUST + BOOKWORM PROJECT TECHNICAL REPORT

DIGITAL HUMANITIES IMPLEMENTATION GRANTS
NATIONAL ENDOWMENT FOR THE HUMANITIES

Project Director: **J. Stephen Downie**
Co-Director, HathiTrust Research Center
Professor & Associate Dean for Research
Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign

Co-Project Director: **Erez Lieberman-Aiden**
Assistant Professor
Baylor College of Medicine

SUBMITTED DECEMBER 7, 2017

Introduction

Bookworm is a tool that visualizes language usage trends at large scales, designed to be powerful but simple. It allows multi-faceted slicing and dicing of the data against a set of content-based and metadata-based features. Our recent work with the HathiTrust+Bookworm (HT+BW) project has focused on improving Bookworm's ability to scale for large collections, while supporting an implementation of Bookworm over one of the largest digital book collections: the HathiTrust Digital Library. The implementation allows scholars to explore the full HathiTrust corpus — but with the control to compare on the basis of such features as subject classification, place of publication, genre, and language. It also provides tools for improved future implementations of Bookworm over non-HathiTrust collections.

Background

The HathiTrust Digital Library is a large-scale digital repository that brings together a corpus of over 15 million volumes of digitized content from more than a dozen partner institutions. This combined corpus provides scholars with a wealth of opportunities for text analysis research. It has seen strong adoption by digital humanists, but the breadth of languages and subjects combined with the depth of the collection makes it very broadly useful.

The HTRC is tasked with supporting exactly this type of scholarly, large-scale usage. It provides the secure computer architecture through which scholars can carry out non-consumptive research. Worksets are a key part of this non-consumptive architecture. They primarily serve as a scholar’s dataset; they are gathered by scholars using a number of methods, including hand curation, database queries submitted to the HathiTrust Digital Library’s vast catalog of metadata records, and even automated sampling algorithms.

The scale of the collection, while certainly its strongest quality, also poses unique obstacles. Making sense of billions of pages is difficult: slow, resource- and skill-intensive, and technically challenging. HT+BW provides solutions to speed up a scholar's early hypothesis-building exploration and to lower the skill barrier to asking complex quantitative questions. It also now provides a method through which scholars can both visualize their worksets and identify new items of interest for inclusion in their worksets.

Beyond the HathiTrust collection, Bookworm also supports custom large collections. The Bookworm documentation (https://bookworm-project.github.io/Docs/) enables even scholars with light technical ability to build their own Bookworms and to showcase some of the features collectively provided by the objects in their worksets. The following sections provide additional details on new features that have been added to Bookworm for scholarly use and the hurdles that had to be overcome to institute them.

Overview

Before continuing, it is useful to understand the scope of Bookworm. There are two components to how a document is represented: the metadata — information about the volumes in the collection — and the data — information about the actual words in the documents.

Fundamentally, a build of Bookworm is a powerful analytic query engine, one that allows you to to ask quantitative questions about the books in the collection conditioned across various metadata facets. One can write a data-based query, such as: "What is the class distribution of books mentioning computers", or a metadata query, such as: "How many books belong to the Library of Congress subclass ‘World War II’, by year?" The underlying engine is accessed through an API (application programming interface) through web access; that is, anyone can structure a query in their browser.

Various tools are available to more easily use the API for statistics or visualization. The most popular tool is a time-series line chart (http://bookworm.htrc.illinois.edu/develop), which can show the frequency with which a word appears in texts over time. This is what many users think of as '*Bookworm,*' because it was the original visualization approach followed by Bookworm and its predecessor, the Google Ngrams Viewer. As it is one of many possible interfaces to Bookworm, we will refer to the time series interface as "Bookworm GUI" here, for clarity. Other available visualization tools are the Bookworm Playground (http://bookworm.htrc.illinois.edu/app), a series of alternate out-of-the-box visualizations, and Bookworm Advanced, a flexible approach towards crafting visualizations and queries together through a declarative grammar. Finally, programmatic access to the API is simplified in Python through the BookwormPython library.

All of these tools connect to the API, and require no privileged access by the HT+BW project. The API is web-accessible and open for cross-domain use. This has two effects on general reuse and access:

1. Scholars may craft their own queries or build their own tools against the entire HT+BW instance. The existing interface tools necessarily make decisions for the user, but scholars with their own unique questions can nevertheless ask them.
2. The tools implemented for HT+BW can be reused for custom, non-HathiTrust implementations of Bookworm.

Access and Use

Here, we detail the forms of use that the HT+BW interfaces support.

Bookworm GUI

The Bookworm GUI allows plotting multiple word trend lines by year. The trends can look across texts, or be specified to be subfacets of the collection. Across the entire collection, the only sensible search comparisons are between different words: e.g. how 'telephone' and 'typewriter' have ebbed and flowed our published works. Through subfacets, however, it is possible to consider identical words in different types of texts: for example, to compare how quickly 'Beijing' was adopted in US books versus British or Canadian books the official transliteration was changed in 1949.

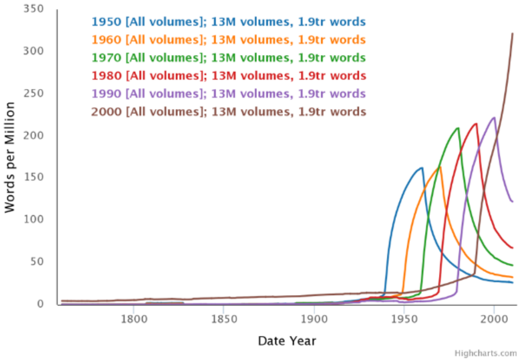


Figure 1: Bookworm GUI query, comparing the occurrence of different year numbers, plotted by year.

The Bookworm GUI always plots information across time. The metric for plotting can be changed from the default, words per million, to percent of all texts, count of all texts, or total count of occurrences. Generally, for comparisons between differently sized subsets of the collection, the relative measures (words per million and text percentage) are more valid.

The facets that can be used for selecting subsets of the data include language, publication country, state, LCC class/subclass/most narrow class, resource type, author name, place, and publisher, among others.

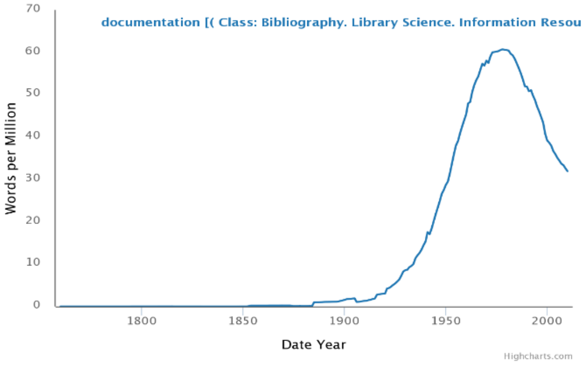


Figure 2: Bookworm GUI query, looking at the rise and fall of 'Documentation' specifically in 184k books classified as *Bibliography, Library Science, and Information Resources*

Example Types of Questions

Verbs over time: Burned v. burnt
Inventions: telephone v. typewriter v. elevator
Colloquialisms: telephone v. phone
Country trends: tea v. coffee in the UK
Subject trends: data / knowledge / information in library science
Censorship: intellectuals in Germany
Narrow class searches: denim v. corduroy in sewing (or fashion)
Changing cultural sensitivities: Eskimo vs. Inuit in US and Canada

The Bookworm GUI is available at http://bookworm.htrc.illinois.edu/develop.

Bookworm Playground

The Bookworm Playground is intended to fill a need between the GUI and advanced interfaces. There is a value to the “quick to learn, quick to use” interface of the time series visualization that makes it much more popular than programmatic or declarative access. However, the latter is much more powerful as a tool for cultural and critical inquiry. The Playground offers user interfaces oriented towards more types of visualizations. While user interfaces require a certain amount of decision-making on behalf of the user, this series of toy tools offers a more extensive sampler of what can be accomplished with Bookworm. The experimental 'playground' branding also makes it a space for rapid deployment, allowing the HT+BW team to publish potentially useful but less polished tools with the user's understanding of that trade-off.
The Playground includes examples of map, heat map, and bar chart interfaces.

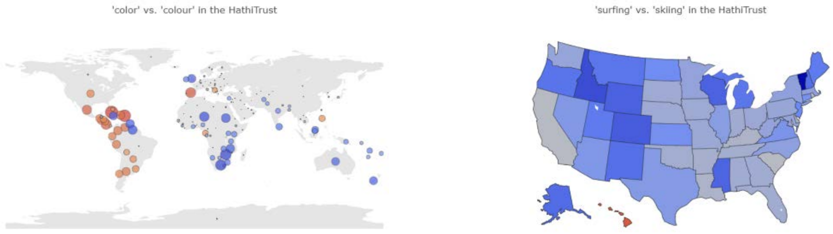

'color' vs. 'colour' in the HathiTrust
'surfing' vs. 'skiing' in the HathiTrust

Figure 3: Map Views

The map visualization (Figure 3 above) allows plotting of word trends by publication country or state, and optionally allows two words to be compared. Selecting a location returns a list of books that contribute to the statistic.

The heat-map visualization plots (Figure 4 below) a search across three dimensions: year (y-axis), words per million (color), and user-selected values from any of the Bookworm facets (x-axis). It is a more elegant alternative to the time-series line charts for instances where there there would be too many lines to compare.

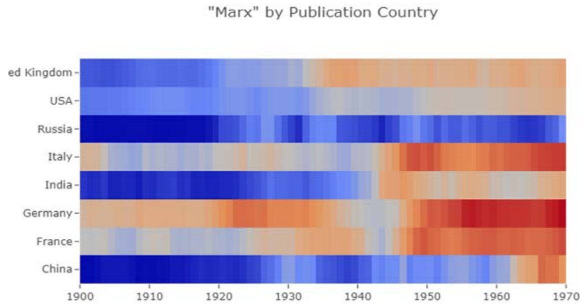
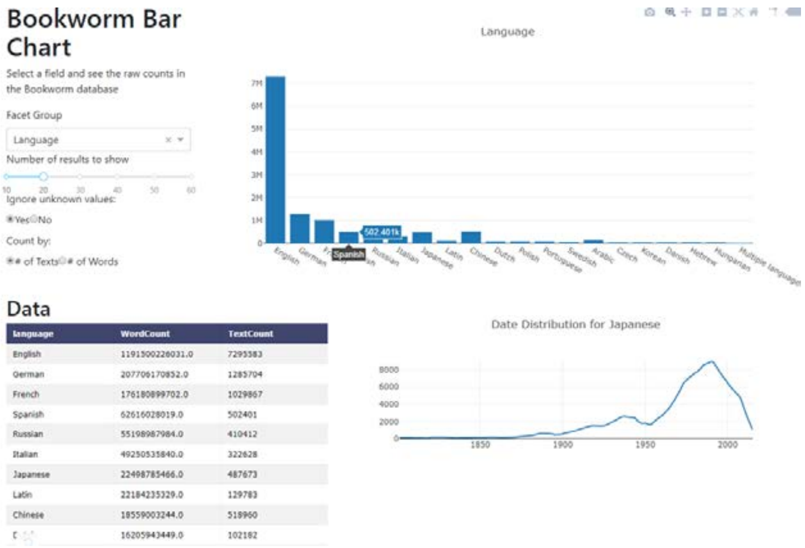


Figure 4: Heatmap in Bookworm Playground

Finally, the bar chart page of the playground offers a dashboard of metadata information. Rather than searching for a word, it provides a glimpse into the distributions of books by facet. For example, in the screenshot below, we see the number of texts per language, the corresponding data in table form, and the date distribution for a selected language.



Bookworm Bar Chart

Select a field and see the raw counts in the Bookworm database

Facet Group: Language

Number of results to show: 20

Ignore unknown values: ● Yes ○ No

Count by: ● # of Texts ○ # of Words

Data

language	WordCount	TextCount
English	1191500226031.0	7295383
German	207706170852.0	1285704
French	176180899702.0	1029867
Spanish	62616028019.0	502401
Russian	55198987984.0	410412
Italian	49250535840.0	322628
Japanese	22498785466.0	487673
Latin	22184235329.0	129783
Chinese	18559003244.0	518960
[illegible]	16205943449.0	102182

Figure 5: Bar chart metadata explorer

The Bookworm Playground is available at https://bookworm.htrc.illinois.edu/app.

Advanced Bookworm Interface

Rounding out visualization tools, the HT+BW project hosts an advanced interface, which uses a declarative visualization grammar to draw various types of data graphics. Consider the following streamgraph, examining how the word 'creativity' groups by class.

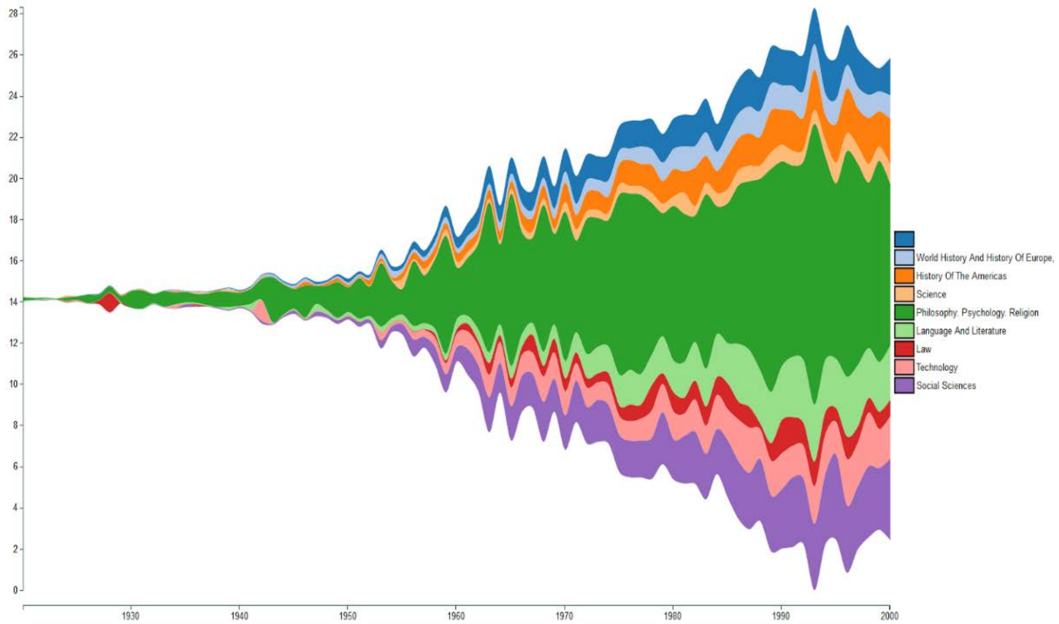
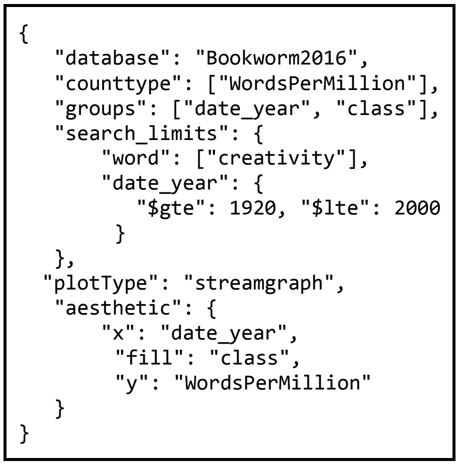


Figure 6: An Example Data Graphic

The associated query to produce the graphic is the following:



```
{
   "database": "Bookworm2016",
   "counttype": ["WordsPerMillion"],
   "groups": ["date_year", "class"],
   "search_limits": {
       "word": ["creativity"],
       "date_year": {
          "$gte": 1920, "$lte": 2000
        }
   },
  "plotType": "streamgraph",
   "aesthetic": {
       "x": "date_year",
        "fill": "class",
        "y": "WordsPerMillion"
   }
}
```

Figure 7: Associated Data Graphic Query

The above query is intended to query the HT+BW database for words per million statistics grouped by year and class, counting the word 'creativity' in texts between 1920 and 2000, and then to plot it to a streamgraph with the x-axis as year, y-axis as count, and fill-color faceted by class.

The advanced interface is an instance of Benjamin Schmidt's Bookworm D3 library..[1] It is available at https://bookworm.htrc.illinois.edu/advanced.

[1] Schmidt, Benjamin. 2015. "Bookworm D3 layouts". http://bookworm.benschmidt.org/posts/2015-10-20-D3-bookworm-plottypes.html

BookwormPython

To aid more advanced use of Bookworm, the HT+BW project developed a Python library for programmatic access to any Bookworm index. It provides scaffolding around the API to assist in using it, validating for errors, and handling the output. One type of output, a Pandas DataFrame, eases the use of Bookworm output to the SciPy stack of data science tools.

BookwormPython is available at https://github.com/organisciak/BookwormPython.

Implementation

The HT+BW implementation focused on key areas related to scale, access, quality, reuse.

- Scale: We focused on improvements to allow more efficient indexing. These activities included developing token-based and compressed input and functionality multi-threaded indexing.
- Access: To support more access vectors to the data, we developed or improved multiple tools, already described above, expanded processes for indexing metadata from MARC records as well as adding arbitrary post-hoc metadata groups. The API was also made accessible from external domains, allowing anybody to query HT+BW.
- Quality: To maximize the quality of HT+BW for inquiry, our activities included the creation of purposive vocabulary whitelists, improved date reconciliation for publications, and corrections for common OCR errors.
- Re-Use: New code was developed in a generalized manner, to keep the relationship distinct between *tool* (Bookworm) and *content* (in this project's case, the HathiTrust corpus). While we focused on the large and notably useful HathiTrust collection, other users can use our code for their own instances of Bookworm.

Metadata

The side-by-side images illustrated by the figure below demonstrate the effects that parsing additional information out of the MARC records have. The curve on the left uses metadata from the publication date field in the MODS metadata records that were originally mined for information. The spikes, especially the large spike for the year 1900, are caused by serial cataloging practices. Generally speaking, neither the dates of individual issues nor those of bound serial volumes are recorded as part of a serial's publication date information. In turn, this causes machine algorithms to assume that the publication date for every volume in a journal's publication run is the same year.

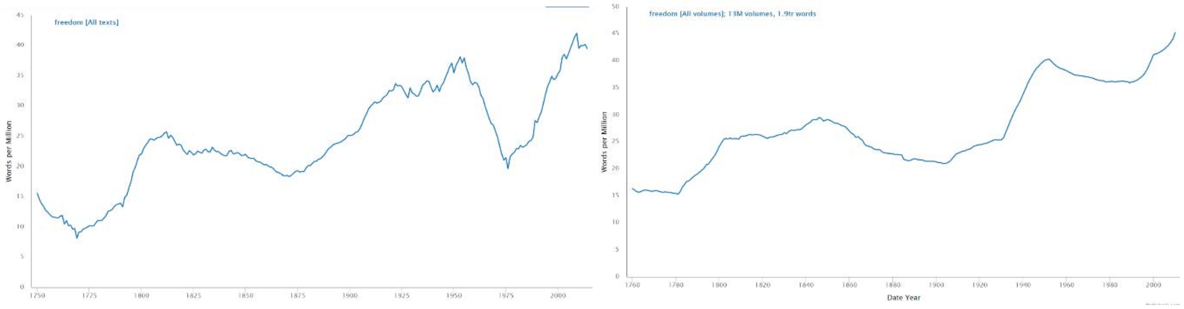


Figure 8: A side-by-side comparison of curves using old (left) and new (right) metadata illustrating the occurrences of the word "freedom" in the HathiTrust corpus.

Some catalogers do record the dates of bound volumes, but this information only appears in a local holdings field, which is not typically parsed by standard metadata parsers. The new algorithms allow the machine to correctly distribute each bound journal volume to its correct year of publication by mining a local holdings field that Zephir— the HathiTrust's's metadata management system (https://www.hathitrust.org/zephyr) run out of the California Digital Library—has been adding to the HathiTrust's MARC metadata records. This in turn results in the smoother, more accurate curve depicted in the image on the right side.

Preparation

HT+BW ingests the latest version of the HTRC's Extracted Features (EF) Dataset and includes a vocabulary of 3,546,302 words generated from the most frequently occurring words in the HathiTrust corpus, which held 13.6 million volumes at the time of development.

The EF Dataset included unigram counts per page for every book in the HathiTrust collection. It holds three particular benefits for HT+BW. First, it reduces the lofty processing overhead of feature extraction, something that was previously done by the EF team on a supercomputer. It also provides cleaned copies of books: re-hyphenation of line-spanning words and identification of undesirable header and footer words has already been performed. Finally, it is an access vector toward the full in-copyright collection. The means the source data for HT+BW is freely available[2] and a scholar can feasibly reproduce our work without privileged access.

The EF Dataset has a few shortcomings. Most notably, HT+BW only supports unigrams, so it cannot be used to research phrases such as 'steam engine' or 'Abraham Lincoln'. Secondly, it is still a piloted dataset, and we discovered that it had tokenization issues with certain Asian languages. While the vast majority of items in the HathiTrust are in European languages (English alone accounts for 49.9% of the corpus), Chinese, Japanese, and Korean texts account for a 6.3% of the corpus. The tokenizers designed for European character sets did not correctly remove the zero-width white-spaces that appear between some pictogram and syllable characters in the three Asian languages, which led to incorrect tokenization by the HTRC's feature extraction algorithms.

We developed a method to fix and fold affected tokens. This resulted in some performance overhead, but hopefully will be unnecessary as the EF Dataset matures.

The code for preparing the EF Dataset is available online, and scholars can prepare other texts for use in the HT+BW pipeline.[3]

Language-Specific Word Lists

For performance, HT+BW cannot support every token in the HathiTrust. At the same time, we would not want to. These texts originated as scanned books, so in addition to rare and uninteresting words, OCR errors are an issue. While the proportion of all words that are errors is very low, the proportion of the *unique words* that are errors is quite high. That is because the space of possible variants for a scanning error is large.

Early in the HT+BW project, we trimmed the vocabulary based on a count of occurrences in the collection: infrequent words were dropped. However, due to the English-language bias of the collection, this strategy unfairly prioritized English over over languages. The long tail of rare and likely erroneous English words would mix with merely infrequent words from other languages.

[2] Boris Capitanu, Ted Underwood, Peter Organisciak, Timothy Cole, Maria Janina Sarol, J. Stephen Downie (2016). *The HathiTrust Research Center Extracted Feature Dataset* (1.0) [Dataset]. HathiTrust Research Center, http://dx.doi.org/10.13012/J8X63JT3.

[3] HTRC Feature Extractor. 2017. https://github.com/htrc/HTRC-FeatureExtractor.

The final solution was a vocabulary comprised of language-specific top-*n* frequency lists. The English language token list was truncated separately from German, Latin, and so on. This required summing the words from five billion pages by language. We anticipate broader public and scholarly value to our language-specific token frequency lists, as well as the final 3.5m word HT+BW whitelist, and we are in the process of dataset documentation for a public release.

Ingest Improvements

The scale of the HathiTrust collection presented multiple challenges for building a Bookworm index. We ultimately addressed it through new indexing features, database parameterization, multi-threading support, compressing input file support, resumable processes, and more thorough logging.

Bookworm indexing typically starts with full text inputs, which are tokenized, summed, filtered, ingested into a database, and finally indexed for performance. The initial full-text tokenization does not work for HT+BW for size and copyright reasons. As already discussed, a pre-tokenized dataset was used instead, and Bookworm was adapted to allow pre-tokenized feature ingest.

While we hope that this improvement will prove valuable for future use, as the project scaled up we found that it was still intractable. Even in feature form, the source data is 1.3 TB compressed, and the two-pass process to count, filter, and uniquely id each word would take a prohibitive amount of time. To address this, a strategy was pursued of working outside of Bookworm's standard processing, allowing us to develop much more performant, multi-threaded (or multi-processor when appropriate) code in a separate environment. The output of this processing was filtered and id-encoded into tables of document/word-count information, compressed in the H5 table format using a fast compression called BLOSC. Again, Bookworm's indexing code was improved to allow this compressed form of input.

Moving processed data into a database, MySQL, created more scale challenges. Data can only be inserted in a single process, and the fastest insert method was found to be MySQL bulk insert of uncompressed text files. Bookworm was enhanced to use multiple processes to decompress parts of input files, while a single thread ingests them into MySQL. It was also made resumable, to also allow partial ingests, and logging was improved to better catch errors early on.

Turning on indexing for database tables – a necessary optimization step – proved to be unwieldy. The details of our solution are beyond the scope of this paper, and are probably particular to specific systems. Broadly speaking, the solution required careful database parameterization, to allow large sorting file sizes and buffer sizes while avoiding system file size limits.

Pedagogical Application

Studies have shown that: a) the complexity of integrating into pedagogical practice text analysis tools operating over large datasets is a significant barrier to their uptake;[4] and b) text analysis tools that seamlessly integrate with the data are a step to overcoming this barrier.[5,6] This motivated our use of HT+BW in the classroom to facilitate exploration the HathiTrust Digital Library's collection without requiring instructors to master complex technology. We used HT+BW in class sessions co-taught by project

[4] Green, H., Dickson, E. and Bhattacharyya, S. (2016). "Scholarly Requirements for Large Scale Text Analysis: A User Needs Assessment by the HathiTrust Research Center." *Digital Humanities 2016* (*DH 2016*), Krakow, Poland. July 2016.

[5] Sinclair, S. and Rockwell, G. (2012). "Teaching Computer-Assisted Text Analysis: Approaches to Learning New Methodologies." In Brett D. Hirsch (ed.), *Digital Humanities Pedagogy: Practices, Principles and Politics*. Cambridge, U.K.: OpenBook Publishers, pp. 241-64

[6] Rockwell, G, Sinclair, S., Ruecker, S. and Organisciak, P. (2010). "Ubiquitous Text Analysis." *paj: The Journal of the Initiative for Digital Humanities, Media, and Culture*, Vol. 2, No. 1.

personnel as part of undergraduate literature classes for students without prior familiarity with quantitative approaches to text analysis.

The goal of the exercises was to help students discover how the meanings of words can vary, in the following cases:

1. Word meanings changing over time; Rens Bod has argued that it is only when humanistic disciplines are compared on a large scale that patterns across them become visible.[7] HT+BW enables the discovery, through active learning, of such patterns by students by investigating trends across categories of knowledge.
2. The same word taking on separate meanings when borrowed from one discipline or domain and applied to a different discipline or domain (or when applied independently in two different domains); this helps students understand how words are often polysemic and/or metaphorical.

Students used HT+BW to generate visualizations consisting of layered time-series plots (stacked area charts) for the relative frequency of their words of interest, within categories of interest in the HathiTrust Digital Library collection. Since the HT+BW tool also provides a subset list of volumes that contribute to the attribute being plotted, students could connect to the actual digitized text of the individual volumes in the list. This affordance of HT+BW helps students bridge the gap between access to a potentially immense corpus and the discovery of specific, relevant individual texts within it that a student can then further investigate through close reading.

Conclusion

The results of the HathiTrust+Bookworm present scholars, students, and citizen scholars with enhanced analytic access to one of the largest digital text corpora, the HathiTrust Digital Library collection, which is an unprecedented aggregation of digitized print materials in hundreds of different languages. The project team was able to overcome many of the challenges represented by limitations with technologies like MySQL, technical formats like the MARC metadata standard, and legal standards like copyright laws. Through the support of the NEH's Office of Digital Humanities for the HathiTrust+Bookworm project (#HK-50176-14) along with the efforts of the project team, it is now possible for scholars and students to explore the billions and billions of words in the HathiTrust Corpus using Bookworm.

[7] Bod, R. (2013). *A New History of the Humanities: The Search for Principles and Patterns from Antiquity to the Present.* New York: Oxford University Press.