Finding Identification Keys in the Biodiversity Heritage Library

by Jorrit H. Poelen, Katja Schulz, Kelli J. Trei and Jonathan A. Rees
10 July 2019

Introduction

Initial Motivation

The Biodiversity Heritage Library (BHL, biodiversitylibrary.org) offers a vast collection of texts to support research and other applications. A number of query tools are available to support exploration of the corpus and discovery of specific works via graphical user and application programming interfaces (see BHL FAQ). However, searching and browsing BHL can still be challenging, and using the full-text (example) or scientific names search (example) often produces long lists of poorly structured search results. In order to improve user experiences, it would be good to enrich BHL metadata with item or page level annotations that provide an index of content elements of interest to BHL audiences, for example:

1. Taxonomic treatments, species descriptions, nomenclatural acts
2. Identification keys
3. Illustrations
4. Plain checklists, seed catalogues, pest reports, etc.
5. References

Specific Goals

For the Global Names/BHL Workshop, we decided to focus on the discovery of identification keys in the BHL corpus, because we expected that these resources would be easy to locate based on their headings and textual structure. In our experience, identification keys are generally preceded by a headline that announces the beginning of the key (e.g., Key to Species, or Key to the Genera of…, or Illustrated Key to North American Species…, etc.). The text of identification keys usually features brief paragraphs with numbers or letters at the beginning of the paragraph, and numbers, letters or taxon names at the end of paragraphs. Series of ellipses often separate the final number, letter, or taxon name from the rest of the paragraph, and illustrations may be interspersed with the key paragraphs.

Proximal goals for the workshop include:

1. Assess whether it is feasible to locate identification keys in the BHL OCR corpus.
2. Explore the utility of regular expressions that rely on the occurrence of the word “key” in the headers of identification keys.
3. Build an index of BHL items that are likely to contain identification keys.
4. Make the index of BHL identification keys available to BHL users.

Potential use cases

1. Finding identification keys for a specific taxon using filter by subject heading in the BHL graphical user interface:

- User goes to biodiversitylibrary.org
- Clicks on search, with full-text option selected
- Searches for a taxon name (e.g., Pseudacris crucifer)
- BHL returns long list of results (>300 publications)
- Current options to limit search results include Type (article, book, etc.), Material (published, archival, maps, computer file, etc), Author, Publication Date, Subject, Language. However, there is no way to specifically search for identification keys.
- Adding "Identification Keys" or something similar to the search fields options would make it easier for users to find keys among search results (Figure 1).

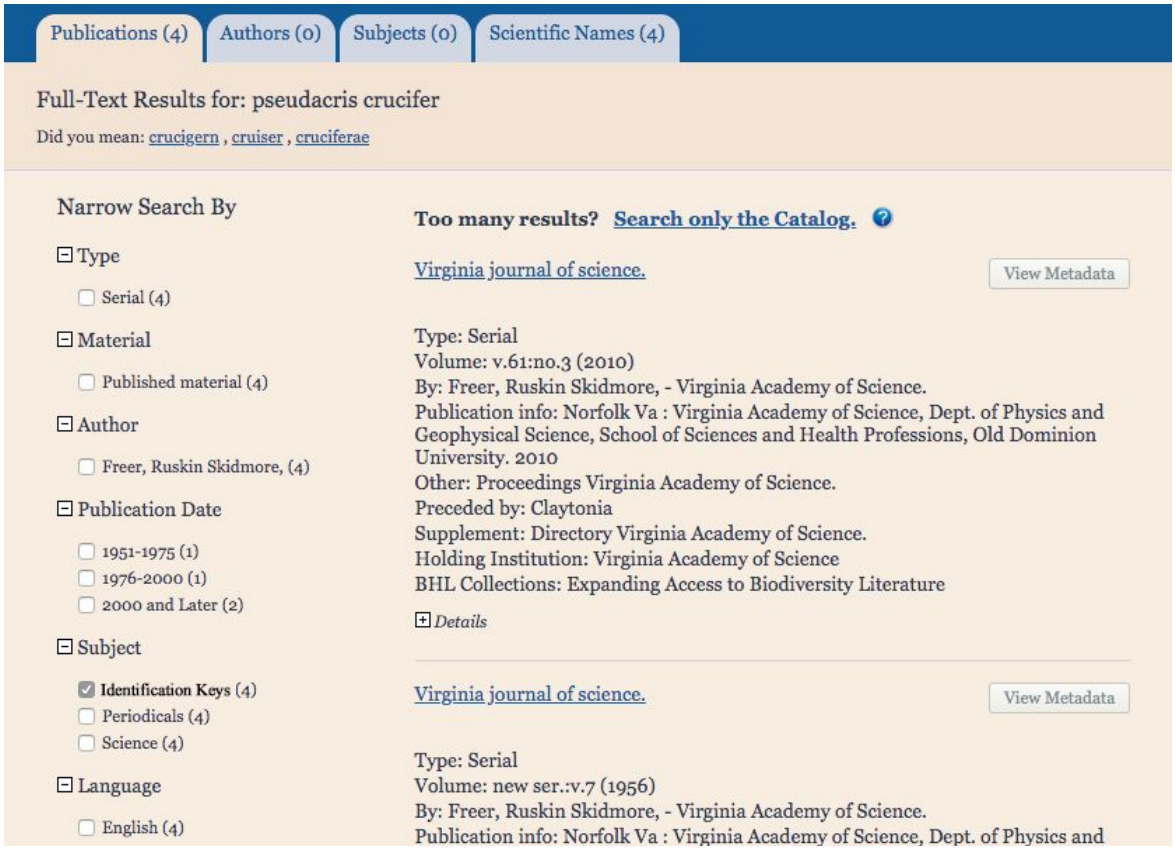


Figure 1: Mockup of the BHL Publications Search interface with the option to filter search results by an example proposed “Identification Key” subject.

- The BHL cataloguing expertise could find related LC classification headings and make appropriate metadata suggestions.
- Adding "Identification Keys" or something similar as a searchable subject would be a good way to integrate links to likely keys into BHL search pages. However, it is not clear what is required to add a Subject heading (e.g. LC classification code) for "Keys" in BHL.

2. Download all items with likely identification keys using the BHL API
    - Users who want to access all BHL items with identification keys could use the Identification Keys subject heading to access all relevant items & download page images or OCR text.
    - It looks like the current BHL API does not support this option. The ***PublicationSearchAdvanced*** method requires that a title, author last name, or collection ID be specified. So it seems there is no way to simply access all publications with a given subject heading.
3. Access items with likely identification keys via curated collection
    - User goes to biodiversitylibrary.org
    - Browse collections
    - Find the curated collection by visual inspection
    - Contents of a given collection can also be accessed through the API, using the ***PublicationSearchAdvanced*** method, e.g.,

https://www.biodiversitylibrary.org/api3?op=PublicationSearchAdvanced&title=&titleop=&authorname=&year=&subject=&language=&collection=53&text=&textop=&page=1&apikey=

- Setting up curated BHL collections for identification keys in works annotated with certain subjects like Entomology, Botany, etc. could be a good first step not requiring any changes to BHL server software.

Also see key_lookup_usecases.md for the initial notes about the use cases.

Methods

Preliminary browsing of identification keys in the BHL corpus (Table 1) revealed that most but not all identification keys in English language works were indeed preceded by headers that mention the word “key,” and most keys used numbers or letters at the beginning of paragraphs. However, we learned that ellipses were not preserved in the OCR text, so they could not be used to locate identification keys in the BHL corpus.

Table 1: Sample of identification keys inspected to explore possible methods for their detection in OCR text.

BHL page id	article title	subject	key header	paragraph elements
16144191	District of Columbia Diptera: Rhagionidae	Entomology	Key to the Species.	numbers
10289299	Bestimmungstabelle für das Zeckengenus Hyalomma Koch.	Entomology	Bestimmungstabelle der Hyalomma♂	numbers
11522797	Analytische Tabelle zur Bestimmung der europäischen Throscus-Arten.	Entomology	Analytische Tabelle zur Bestimmung der europäischen Throscus-Arten.	numbers
35918404	Tabelle zur Bestimmung der Tanythrix-Arten.	Entomology	Tabelle zur Bestimmung der Tanythrix-Arten.	letters
10435324	Bestimmungstabellen von Insekten-Larven.	Entomology	Tabelle zur Bestimmung der Necrodes- und Silpha-Larven.	none

55440554	A revised identification guide to the fairy shrimps (Crustacea: Anostraca: Anostracina) of Australia	Carcinology	Key to families and genera of Anostraca in Australia	numbers & letters
36251191	A checklist of Canadian Atlantic fishes with keys for identification	Ichthyology	Key to Order PLEUROTREMATA - sharks	numbers
48287659	Field Guide to the Amphibians of Western India	Herpetology	Key to the Genera in Western India of the Family Ranidae	numbers
47008116	Key to North American Birds	Ornithology	KEY TO THE GENERA.	letters
48422132	Identification of the larger fungi	Mycology	Key to major genera	numbers
3533145	Flora of Illinois	Botany	KEYS TO GENERA AND SPECIES	numbers
4945798	Woody Plants of the Western National Parks	Botany	Field Guide to the Trees	none
48287225	A new species and notes on the genus Anthoxanthum L. (Poaceae)	Botany	Key to species	numbers

As a first step, we decided to stick to an item-level approach that would allow us to create an index of BHL works that contain identification keys. Ideally this would be followed with a more precise approach that would pinpoint the specific locations of keys in the text, so we could support queries of taxon names that occur in identification keys.

The following procedure was developed to iteratively improve a method to detect likely identification keys in biodiversity texts based on the occurrence of the word “key”:

1. Get the fully available BHL data/corpus
2. Assess the completeness of the corpus
3. Find lines that may indicate the beginning of identification keys in the corpus

4. Save results in a file with (at least) three columns: (1) Extracted matching line with "key" occurrence and (2) item id (3) line number of "key" occurrence
5. Domain expert visually inspects matching lines for false positives
6. When needed, the expert reviews the original text containing the matching line
7. Refine regular expression to work around false positives.

Acquisition of BHL corpus

The BHL corpus was accessed via a Preston BHL archive. The archive used consisted of two versioned snapshots of the BHL corpus obtained over the period of period May-June 2019. Each snapshot contains item level OCR texts as well as a detailed record from which url the OCR texts were obtained and when. Each obtained OCR text is identified by its content hash. A content hash is an algorithmically generated unique identifier based on, and only on, the content of the OCR text. In case of Preston, sha256 content hashes are used (see https://preston.guoda.bio). This detailed information helps to establish a link between the content that was present at a specific time to a BHL item identifier. By using the Preston BHL archive, we can link OCR text to its BHL item identifier in addition to uniquely identifying what OCR text was used. In addition to this, the archive contains a record of missing OCR texts, or OCR texts that could not be accessed at the expected location in the internet archive.

The BHL archive was accessed using an external harddisk to optimize data retrieval. This archive was retrieved from a remote server location (e.g., https://deeplinker.bio) using rsync (Andrew Tridgell, Paul Mackerras. 1998. The Rsync Algorithm. Accessed on 2019-06-28 at https://rsync.samba.org/tech_report/) prior to the workshop. The BHL archive total size was 120GB and consisted of 227k OCR texts and a single tabular file containing the BHL item catalog. The item catalog is updated weekly (Pers. Comm. Mike Lichtenberg/Joel Richard from BHL) and contains all items in BHL. The BHL archive used can also be retrieved from Poelen, Jorrit H. (2019). A biodiversity dataset graph: Biodiversity Heritage Library (BHL) (Version 0.0.1) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.3251134 and https://archive.org/details/preston-bhl with provenance hash://sha256/41b19aa9456fc709de1d09d7a59c87253bc1f86b68289024b7320cef78b3e3a4 .

Assessment of completeness

To assess the completeness of the versioned BHL corpus archive used for this prototype, the available OCR texts were linked to the BHL item catalog. This showed that out of 285k items, 57k items, or 20%, were missing OCR texts. For example, the following 10 urls failed to serve OCR texts:

```
$ cat bhl_djvu_404.tsv | head
https://archive.org/download/00921238.85096.emory.edu/00921238.85096.emory.edu_djvu.t
xt
https://archive.org/download/02145706.5485.emory.edu/02145706.5485.emory.edu_djvu.txt
```

```
https://archive.org/download/0220434.nlm.nih.gov/0220434.nlm.nih.gov_djvu.txt
https://archive.org/download/03060843.1594.emory.edu/03060843.1594.emory.edu_djvu.txt
https://archive.org/download/03060843.1595.emory.edu/03060843.1595.emory.edu_djvu.txt
https://archive.org/download/03060843.1596.emory.edu/03060843.1596.emory.edu_djvu.txt
https://archive.org/download/03060843.1597.emory.edu/03060843.1597.emory.edu_djvu.txt
https://archive.org/download/03060843.1598.emory.edu/03060843.1598.emory.edu_djvu.txt
https://archive.org/download/03060843.1599.emory.edu/03060843.1599.emory.edu_djvu.txt
https://archive.org/download/03060843.1600.emory.edu/03060843.1600.emory.edu_djvu.txt
```

The full list of broken urls links can be found at bhl_djvu_404.tsv. Root causes for missing OCR texts appear include: (1) OCR text are stored in a non-standard file (e.g., https://www.archive.org/download/McGillLibrary-129682-5040), and, (2) the OCR text failed to be generated from the dvju.xml file (e.g., https://archive.org/download/mobot31753000810538). For more details related to referenced examples of missing OCR texts, please see Appendix B.

Finding identification keys

Script for the detection of key candidates

After acquiring the BHL corpus and establishing that our BHL corpus contained approximately 80% of the available OCR texts, a bash script, find_keys.sh was created to match all lines in OCR text against a regular expression. The script takes two arguments (1) the location of the BHL corpus and (2) a regular expression for matching likely keys.

The script was executed using the BHL corpus on an external hardisk attached to a dual core lenovo T430 laptop with 8GB of memory using regular expression `\bkey\b`. It took about 50 minutes for the script to match all the lines of the OCR text. The output of the script captured the line number, text, content hash and related BHL, internet archive, and Preston urls in the file itemurl-line-match.tsv. The file contains 758k unique matches, including the 10 lines in Table 2. Note that the urls were turned into labels ia (internet archive), bhl (biodiversity heritage library) and OCR test (a url pointing to a preston archive at https://deeplinker.bio) to declutter the table representation.

Table 2: A sample of the lines from the output of the find_keys.sh script.

item links	line number	matching line
ia bhl ocr-text	10867	belong we have the key-note to the common
ia bhl ocr-text	11012	living beings, it gave him the key to many mys-

ia bhl ocr-text	12431	common Five-Finger (Asterias) gives the key to
ia bhl ocr-text	2616	and is the key to their whole organization. A
ia bhl ocr-text	3049	fications ; and that we have already the key by
ia bhl ocr-text	513	whole. It was Cuvier who found the key. He
ia bhl ocr-text	5306	pitched on a different key, it is true, but a sound
ia bhl ocr-text	673	and gave us the key-note to the natural affinities
ia bhl ocr-text	7270	crescent, from Virginia Key and Key Biscayne,
ia bhl ocr-text	7271	almost adjoining the main-land, to Key West, at

First pass at assessing occurrence of “key” in the BHL corpus

Since the script for the detection of key candidates was not ready by the second day of the workshop, we developed the following query with the assistance of Joel Richard to extract all lines in the BHL corpus that mention the work “key”:

```
grep -r -n -i -e '\bkey\b' data/ > BHLKeySamples.tsv
```

We chose to begin with this very general query in order to assess the general usage of the word key within the item level text. This resulted in a file (see BHLKeySamples.tsv.zip) with 3 columns (filepath, line number, extracted line) and 770114 lines that could be used in a preliminary manual assessment of the use of the word “key” in BHL texts. To facilitate access to the context of each line in the original works, file paths were converted to https://deeplinker.bio/ links (see BHLKeySamplesDeeplinker.tsv.zip).

Results

Analysis of the find_keys.sh script output conveyed that 75k items (32% of items with OCR text) have one or more matches to `\bkey\b`. Also, there are few items with many counts (e.g. approximately 100 items with 500 matches or more) and many items with few counts (i.e., about 65k items with 10 matches or less, Figure 2).

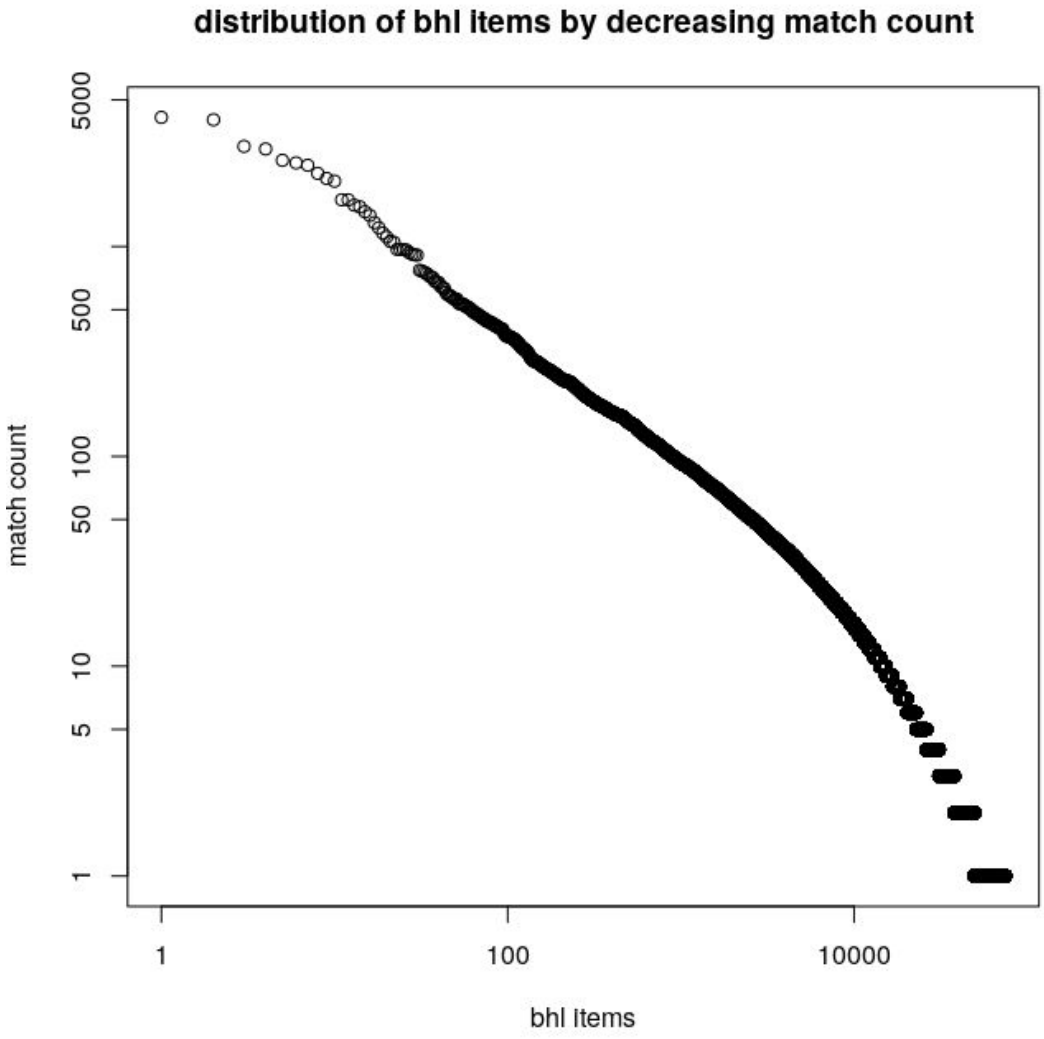


Figure 2: Number of occurrences of lines matching the `\bkey\b` query across items in the BHL corpus.

Manual inspection of the items with the highest match count revealed that they were all entomological catalogues with an abundance of references to works containing keys (Table 3). Number of match count therefore does not appear to provide a means to locate likely keys in the corpus.

Table 3: The 10 items with the highest number of lines matching the `\bkey\b` query. Manual inspection revealed that none of these items contained any identification keys.

#key lines	item link	item title	keys in item?
4127	107146	General catalogue of the Homoptera: Cicadelloidea: Cicadellidae	no

4020	118981	A catalogue and reclassification of the Nearctic Ichneumonidae	no
3002	262128	Catalogo dos Mirideos do Mundo I	no
2916	107123	General catalogue of the Homoptera: Cicadelloidea: Euscelidae I	no
2576	107124	General catalogue of the Homoptera: Cicadelloidea: Euscelidae II	no
2505	194866	General catalogue of the Homoptera: Membracoidea I	no
2440	261069	Catalogo dos Mirideos do Mundo II	no
2231	194618	General catalogue of the Homoptera: Membracoidea II	no
2111	107164	General catalogue of the Homoptera: Cicadelloidea: Tettigellidae	no
2049	29885	General catalogue of the Hemiptera: Fulgoroidea III	no

For more details on the results, please refer to Appendix A. This appendix contains code snippets and result logs that help to reproduce the results provided that the same version of the BHL archive is used. If you decide to do that, it's possible to algorithmically verify that you are using the same Preston BHL archive that we did.

Initial inspection of the first pass results revealed that many of the 770114 lines did not map to headers for identification keys. The word key is used in a variety of contexts that are not related to identification resources. Some of the non-relevant lines could be excluded based on one of the following rationales:

1. Lines containing only lowercase letters, indicating that they represent portions of longer sentences rather than headlines. – 119312 lines.
2. References, recognized as lines containing 4-digit numbers starting with 18,19, or 20. – 82903 lines.
3. Putative placenames, e.g., Key Largo, Pelican Key. US placename data were downloaded from Geographic Names Information System (GNIS) via the National File,

international placename data were downloaded from GEOnet Names Server (GNS) via a Name and Designation Request. Placenames were extracted, deduplicated and manually cleaned. – 73789 lines.

4. Common word groups that are unlikely to be part of the header of an identification key: “key word” (27552 lines), “lock and key” (2424 lines), “key role” (2045 lines), “key factor” (1834 lines), “key feature” (1344 lines), “key to success” (1242 lines), “key element” (1209 lines), “key component” (848 lines), “key to abbreviations” (689 lines), “key to symbols” (507 lines), “key concept” (230 lines).
5. Mentions of keys in taxonomic catalogues, usually in the form of [key] or [in key] or (key) or (in key). – 42572 lines.

Removing these lines results in a smaller data set with 438243 lines that still contains many non-relevant lines based on manual inspection.

Headers of identification keys that were confirmed through reference to the original work were often simple (e.g., “Key to Species”) but could also be complex containing a variety of **prefixes** (e.g., determination key, draft key, field key, identification key, master key, multi-access key) and/or **adjectives** (e.g., analytic, analytical, annotated, artificial, brief, complete, composite, comprehensive, concise, descriptive, detailed, diagnostic, dichotomic, dichotomous, general, generic, illustrated, improved, modified, morphological, new, partial, pictorial, preliminary, provisional, revised, short, simple, simplified, synoptic, synoptical, systematic, tabular, taxonomic, tentative, updated, visual, working) to describe the identification resource. Information about the **geographic** (e. g., African, Afrotropical, American, antarctic, Asiatic, Atlantic, Australasian, Australian, British, Central American, Chinese, Colombian, Eastern, Egyptian, European, Indian, Nearctic, Neotropical, North American, Oriental, Pacific, Palaearctic, South African, South American, Swedish, tropical, Western, world, worldwide), **ecological** (aquatic, benthic, deep-sea, estuarine, freshwater, land, marine, parasitic, pelagic, soil-inhabiting, terrestrial), **taxonomic** (e. g., genera, families, family, order, species, spp., subfamilies, tribes, various taxon names) and/or **life history/reproductive** (e.g., adult, eggs, females, fruiting, genitalic, hermaphrodite, immature, juveniles, larvae, larval, males, nymphs, pupae, sterile, vegetative) scope of the key may also be included in the header.

Discussion

The BHL corpus presents many opportunities for discovery and analysis ranging from simple regular expression matching to subsection classification (abstract, identification keys) to information extraction (e.g. trait information). Locating identification keys in the BHL corpus proved to be more difficult than expected. The major problem is that we are unable to develop a precise method to distinguish the headings and subheadings introducing a key from citations of works that are identification keys and from other mentions of keys that are not immediately followed by a key. Headings are not distinguished in the current OCR text, and special

characters (e.g., ellipses) that could serve as landmarks for putative keys are lost, so proximity searching for these near titles is not an option. Since key titles often appear in references, even matching the exact title is not helpful in identifying the beginning of a key.

One of the major obstacles to a more thorough exploration of our preliminary result set was the lack of an efficient method for accessing the original context of extracted "key" lines. While we had links to the original OCR text files and the line numbers for the extracted text, investigating the context of each line required downloading the text files and manually navigating to the relevant page number. Having a tool available that would support click and go access directly to a specific line number in a particular text file would greatly improve the context exploration workflow.

Because identification keys generally have a distinct structure, the application of machine learning algorithms may be useful for locating keys in the BHL corpus. These could be trained either on OCR text or on page images. To assess the feasibility of this approach, we need to explore software libraries or platforms that can help with text classification (e.g., does this text contain a key?) and segmentation (e.g., which part of the text is a key)? (Stanford CoreNLP, OpenNLP, http://brat.nlplab.org). Tools for the efficient annotation and selecting of OCR and/or page image text are also needed to facilitate the creation of a diverse training set of keys and key near misses from the BHL corpus.

The BHL corpus is large, about 120G of retrieved OCR output at the time of this project, and it is desirable to be able to do analysis tasks reasonably quickly. Four factors contribute to overall execution time:

1. The speed of processing individual files (pages or items) using file-level programs like `grep`
2. Parallelism due to pipeline streaming - sequential stages can run in parallel on different parts of one file
3. Parallelism due to concurrent pipeline execution ('scatter-gather' or 'map-reduce')
4. Disk IO - processing delay due to hard disk lookup up a file and reading the contents

Fortunately, many useful corpus operations can be characterized as stream or pipeline processing on a large number of independent files, and therefore highly parallelizable. For example, scientific name matching, which has relatively high per-file processing cost, can be completed in five hours thanks to the high parallelism available on an ordinary computing cluster. [reference: Dmitry Mozzherin communication during this workshop]

To produce an expository prototype in a short amount of time, we wanted to choose tools that were known to both of the developers (JP and JR). The primary tool ended up being `bash`, together with the classic Unix command line utilities. `bash` assists with pipeline parallelism

but not with scatter-gather. In spite of this limitation, our `bash` script was able to scan the entire corpus in less than an hour (see results, above) even though it processed the corpus sequentially.

The other important tool for this job was `preston` for managing the BHL download and versioning, written by JP. `Preston` was simple enough that JR could learn it sufficiently during the workshop.

Steps in the task:

1. Obtain BHL corpus (item OCR files only) from the Internet Archive using `preston`
2. Scanning all item OCR files for matches to a regular expression, obtaining a table with local file name, line number, and matched text
3. Annotate matches with URLs related to the corresponding BHL item:
    a. Internet Archive link based on 'barcode' identifier;
    b. deeplinker.bio link based on hash of the file, replacing the local file name also based on hash;
    c. BHL item link based on item id

Producing this script was a typical software project in that we thought we'd finish in about 30 minutes, and it ended up taking about five hours. Most of this time was spent wrestling with the `sed` command line utility for transforming strings, which we needed in order to change field separators and prefixes on identifiers. Sed fails silently on any kind of error. We had to get used to group replacement operators, character escaping, and various other awful details of `sed` regular expressions, which differ in subtle ways from other regular expression languages.

For BHL OCR file harvesting and versioning, `preston` worked quite well. It is similar in ways to large-object extensions to git, but is coupled with a web harvesting system that allows controlled versioned update of resources loaded from the web.
For rapid turnaround in script development we tested against a test corpus that was a small subset of the entire BHL corpus.

Advice

Based on our experience we have some recommendations for anyone trying something like this in the future:

Write and use unit tests.

Consider using a programming language that has support for string transformations, especially one that has a unit test framework, even if the script ends up being more verbose than as `bash/sed` script would be. Any of the usual suspects (perl, php, python, ruby, java) would

probably be better than a shell such as `bash`, if only because their regular expressions are usually less idiosyncratic than `sed`'s and therefore more familiar to most programmers.

The disadvantages of these languages compared to bash is in clumsier invocation of programs such as grep, and clumsier support (or no support at all) for processing pipelines.

Consider use of some framework for parallelization. We talked about but did not analyze the following options:

- Go (golang) - this is of interest based on Dima's testimony regarding its ability to use all available compute resources in a computing cluster. Dima also has positive things to say about kubernetes, which he uses for deployment and resource management.
- Scala
- Spark
- Workflow systems e.g. Taverna
- `make -j`, which is a very low-tech method that suppresses work that has already been completed on a previous partial run

Appendix A

This appendix contains logs and code snippets to help reproduce the results.

The listing below contains the log of the execution of the find_keys.sh scripts.

```
$ time ./find_keys.sh /media/jorrit/cobaltblue/preston-bhl/ "\bkey\b"
+ OLD_PWD=/home/jorrit/proj/gn-hackathon/Identification-Key-Finder
+ DATA_DIR=/media/jorrit/cobaltblue/preston-bhl/
+ REGEX='\bkey\b'
+ cd /media/jorrit/cobaltblue/preston-bhl/
+ echo 'Scanning BHL corpus for matches to regular expression'
Scanning BHL corpus for matches to regular expression
+ echo 'output to
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/hash-line-match.tsv'
output to
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/hash-line-match.tsv
+ find data -type f
+ xargs grep -n '\bkey\b' -w -i
+ sed -e 's+^.*/../../++'
+ sed -e 's/:/\t/'
+ sed -e 's/:/\t/'
+ cat
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/hash-line-match-unsorted.tsv
+ sort
+ echo 'Obtaining map from file hash to file barcode'
```

```
Obtaining map from file hash to file barcode
+ preston log -l tsv
+ cat /home/jorrit/proj/gn-hackathon/Identification-Key-Finder/log.tsv
+ grep 'archive.*hasVersion'
+ grep -v well-known
+ sed -e 's+[ia](https://archive.org/download/\([^/]*\).*sha256/\(.*\)+\2\t\1+'
+ sort
+ echo 'Obtaining map from barcode to BHL item'
Obtaining map from barcode to BHL item
+ cat /home/jorrit/proj/gn-hackathon/Identification-Key-Finder/log.tsv
+ head -n100
+ head -n 1
+ grep hasVersion
+ grep item.txt
+ tail -n +2
+ sort
+ cut -f8,4
+ cut -f 3
+ preston get
+ echo 'Joining on file hash and sorting'
Joining on file hash and sorting
+ join --nocheck-order -t '      ' -1 1 -2 1
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/hash-barcode.tsv
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/hash-line-match.tsv
+ sort -k 2
+ echo 'Joining on barcode and sorting'
Joining on barcode and sorting
+ join --nocheck-order -t '      ' -1 1 -2 2
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/barcode-itemurl.tsv
/home/jorrit/proj/gn-hackathon/Identification-Key-Finder/barcode-line-match.tsv
+ sed -e
's+\(.*\)\t\(.*\)\t\(.*\)\t\(.*\)\t\(.*\)+[ia](https://archive.org/download/\1\t\2\t[
ocr-text](https://deeplinker/\3\t\4\t\5+'

real   46m36.495s
user   9m22.076s
sys    3m57.117s
```

Result Statistics

Counting unique number of line matches again BHL corpus using regex `\bkey\b`.

```
$ zcat itemurl-line-match.tsv.gz | sort | uniq | wc -l```)
758180
```

Calculating the number of unique BHL items with matches.

```
$ zcat itemurl-line-match.tsv.gz | sort | uniq | cut -f2 | sort | uniq | wc -l```)
74719
```

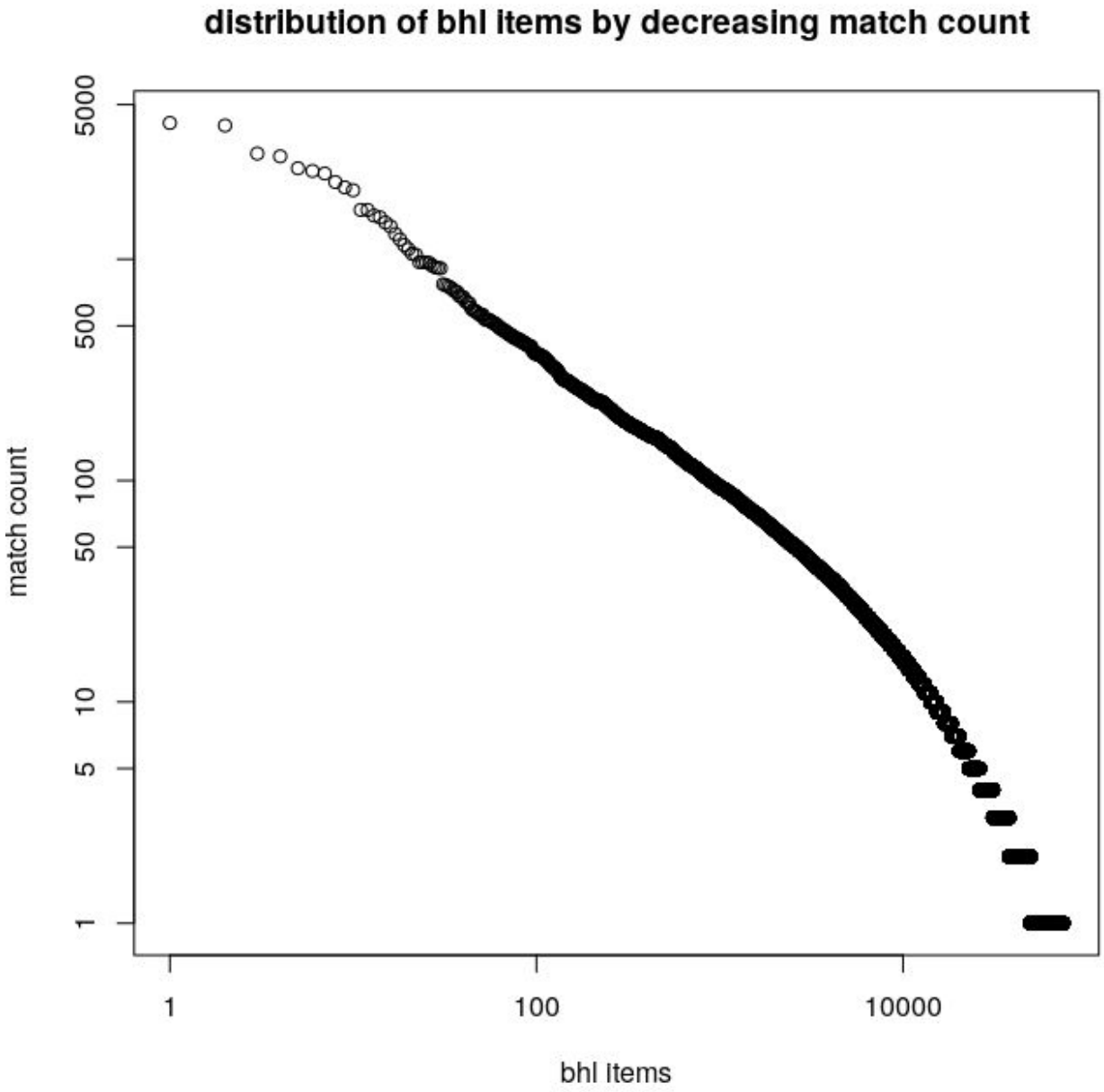
Calculating the top 10 BHL items with most number of matches.

```
$ zcat itemurl-line-match.tsv.gz | sort | uniq | cut -f2 | sort | uniq -c | sort -nr
\
    | sed "s/^[^0-9]*//g" | tr ' ' '\t' \
    > matches-per-item-sorted-descending.tsv
$ head matches-per-item-sorted-descending.tsv
4127   https://www.biodiversitylibrary.org/item/107146
4020   https://www.biodiversitylibrary.org/item/118981
3002   https://www.biodiversitylibrary.org/item/262128
2916   https://www.biodiversitylibrary.org/item/107123
2576   https://www.biodiversitylibrary.org/item/107124
2505   https://www.biodiversitylibrary.org/item/194866
2440   https://www.biodiversitylibrary.org/item/261069
2231   https://www.biodiversitylibrary.org/item/194618
2111   https://www.biodiversitylibrary.org/item/107164
2049   https://www.biodiversitylibrary.org/item/29885
```

Constructing a figure with a distribution of number of matches across BHL items with at least one match using R (see https://r-project.org).

```
plot(item_matches$count, log="xy")
item_matches <- read.csv('matches-per-item-sorted-descending.tsv', header=F,
sep='\t')
names(item_matches) <- c('count', 'item_url')
plot(item_matches$count, log="xy", xlab='bhl items', ylab='match count',
main='distribution of bhl items by decreasing match count')
```

The R script above was used to produce the plot below.



system info

A Ubuntu Linux 18.04 operating was used, running on Lenovo Laptop T430 8GB RAM with dual core Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz.

BHL Corpus

Calculate number of items in the BHL catalogue:

```
$ cat item.txt | wc -l
242511
```

Note that the file item.txt was extracted from the Preston BHL archive used.

Total number of files in the Preston BHL archive:

```
$ find data -type f | wc -l
226900
```

Total volume of Preston BHL archive files:

```
$ du -d0 -h data/
120G   data/
```

Preston BHL Archive version:

```
$ preston version
0.0.15
$ preston history
<0659a54f-b713-4f86-a917-5be166a14110> <http://purl.org/pav/hasVersion>
<hash://sha256/89926f33157c0ef057b6de73f6c8be0060353887b47db251bfd28222f2fd801a> .
<hash://sha256/41b19aa9456fc709de1d09d7a59c87253bc1f86b68289024b7320cef78b3e3a4>
<http://purl.org/pav/previousVersion>
<hash://sha256/89926f33157c0ef057b6de73f6c8be0060353887b47db251bfd28222f2fd801a> .
```

Appendix B

This appendix examples of missing OCR text pages.

Example 1. McGillLibrary-129682-5040

from https://www.biodiversitylibrary.org/data/item.txt (see also hash://sha256/e0c131ebf6ad2dce71ab9a10aa116dcedb219ae4539f9e5bf0e57b84f51f22ca)

```
$ preston get hash://sha256/e0c131ebf6ad2dce71ab9a10aa116dcedb219ae4539f9e5bf0e57b84f51f22ca | grep "McGillLibrary-129682-5040"

195710 112053 49506791     McGillLibrary-129682-5040 McGillLibrary-129682-5040       https://www.biodiversitylibrary.org/item/195710 McGill University Library (archive.org)       2016-01-24 00:38
```

was unable to find

https://www.archive.org/download/McGillLibrary-129682-5040/McGillLibrary-129682-5040_djvu.txt

Note that https://archive.org/download/McGillLibrary-129682-5040/ contains the non-standard file "129682_djvu.txt", whereas the expected file "McGillLibrary-129682-5040_djvu.txt" is missing.

See also,

https://www.biodiversitylibrary.org/item/195710

download content -> download book -> download text -> 404 not found

Example 2. mobot31753000810538

Similarly,

```
$ preston get hash://sha256/e0c131ebf6ad2dce71ab9a10aa116dcedb219ae4539f9e5bf0e57b84f51f22ca | grep "mobot31753000810538"
```

```
14549  724    529604 mobot31753000810538 i1269521x    QK41 .C57 1601 [#1006]
https://www.biodiversitylibrary.org/item/14549             1601   Missouri Botanical
Garden, Peter H. Raven Library         2006-05-04 00:00
```

```
$ preston ls | grep mobot31753000810538

<hash://sha256/e0c131ebf6ad2dce71ab9a10aa116dcedb219ae4539f9e5bf0e57b84f51f22ca>
<http://www.w3.org/ns/prov#hadMember> <mobot31753000810538> .

<mobot31753000810538> <http://www.w3.org/1999/02/22-rdf-syntax-ns#seeAlso>
<https://archive.org/download/mobot31753000810538/mobot31753000810538_djvu.txt> .

<https://archive.org/download/mobot31753000810538/mobot31753000810538_djvu.txt>
<http://purl.org/dc/elements/1.1/format> "text/plain;charset=UTF-8" .

<https://archive.org/download/mobot31753000810538/mobot31753000810538_djvu.txt>
<http://purl.org/pav/hasVersion>
<https://deeplinker.bio/.well-known/genid/40662b2b-a402-3f32-9cce-f225c07d2d00> .
```

where

```
<https://archive.org/download/mobot31753000810538/mobot31753000810538_djvu.txt>
<http://purl.org/pav/hasVersion>
<https://deeplinker.bio/.well-known/genid/40662b2b-a402-3f32-9cce-f225c07d2d00>
```

means that

https://archive.org/download/mobot31753000810538/mobot31753000810538_djvu.txt

could not be accessed.

Note that the file djvu xml file "mobot31753000810538_djvu.xml" exists, but the associated text file "mobot31753000810538_djvu.text" does not.

https://www.biodiversitylibrary.org/item/14549

download content -> download book -> download text -> 404 not found