BRILL

RESEARCH DATA JOURNAL FOR THE HUMANITIES AND SOCIAL SCIENCES (2024) 1–21

BRILL OPEN

brill.com/rdj

The Brill Knowledge Graph: A Database of Bibliographic References and Index Terms Extracted from Books in Humanities and Social Sciences

Natallia Kokash
Institute for Logic, Language and Computation,
University of Amsterdam, Amsterdam, the Netherlands
natallia.kokash@gmail.com

Matteo Romanello
Institute of Archeology and Classical Studies,
University of Lausanne, Lausanne, Switzerland
matteo.romanello@unil.ch

Ernest Suyver
Brill, Leiden, the Netherlands
ernestsuyver@gmail.com

Giovanni Colavizza | ORCID: 0000-0002-9806-084X
Corresponding author
University of Amsterdam, Amsterdam, the Netherlands
Department of Classical and Italian Philology,
University of Bologna, Bologna, Italy
giovanni.colavizza@unibo.it

Received 15 March 2023 | Revised 22 December 2023 |
Accepted 19 February 2024 | Published online 24 April 2024

Abstract

With the recognition of content discoverability and information analytics in the scientific publishing industry, more and more effort is dedicated to digitization and automated analysis of scholarly publications. As part of this effort, the authors

designed a pipeline to extract structured information from bibliography and index lists of existing scholarly publications, as well as to disambiguate and export it as linked data. In this article, the authors present the Brill Knowledge Graph (KG), obtained by applying this pipeline to a corpus of books in the Arts, Humanities and Social Sciences (AHSS) provided by the publisher Brill.

Keywords

data mining – PDF parsing – bibliography – index – knowledge graph – academic publishing

– Related data set "The Brill Knowledge Graph: A Database of Bibliographic References and Index Terms extracted from Books in Humanities and Social Sciences" with DOI www.doi.org/10.5281/zenodo.7691771 in repository "Zenodo"

1. Introduction

Recently there has been an increased interest in citation databases and related online services facilitating the discovery of scholarly publications (Naik & Pai, 2020; Rahm & Thor, 2005). Citation databases are collections of referenced papers, articles, research reports, manuscripts, books and other materials represented in a structured and consistent way. Such databases enable targeted search requests that can be constrained, for example, by author or title fields, or year of publication, and keywords and index terms can be used to locate works dedicated to a certain subject. Searches undertaken in citation databases are typically more precise and comprehensive than searches on general internet search engines (Linder et al., 2015).

There is a considerable opportunity to improve the quality and coverage of citation databases. Many database records are still handled manually (Nazarovets, 2021). What is more, the works produced before the mainstream digitization of the publishing process are often not covered. In some fields such as Arts, Humanities and Social Sciences (AHSS), uneven citation indexation is more pronounced in part due to the prevalence of small and medium publishers who do not have resources to maintain an integrated citation database.

Motivated by the aforementioned issues, we designed and implemented an open-source data mining pipeline to help small and medium publishers to automatically extract citation and index data from existing publications

(Kokash et al., 2023a). This article presents a complete dataset of linked bibliography and index data, partially disambiguated and augmented with references to external resources, extracted from Brill's archive in the field of Classics. The dataset is constructed by our data extraction pipeline which locates back matter files in PDF format based on the publisher's JATS file annotations and splices them into lists of bibliographic references and index terms. Text fragments extracted from different books via this process are then parsed and compared using a string-based similarity metric (Prasetya et al., 2018) to form clusters of bibliographic references to the same published work or (variants of) the same subjects discussed in these books. The entire set of references was then disambiguated using Google Books (Google Books APIs, 2012] and Crossref APIs (Crossref REST API, 2016). We refer the reader to Kokash et al. (2023a) for further methodological details on this workflow.

2. Data Extraction

The source data used for the creation of this database consists of books in the field of Classics provided by the publisher Brill. The corpus includes 1816 books produced in the period 2006–2021. We estimated that the dataset includes 965 edited volumes (collections of chapters by different authors), and 851 monographs (books on one subject written by a single author), with an average of 369 pages per book. Books are archived in compressed folders which include PDF files with content (one file per each chapter and a file with full text), back matter (one or more files with bibliographic references and indexes) and a metadata file in BITS XML format that outlines the book structure. All book annotations are available at the publisher's website (www.brill.com). We compiled a list of Digital Object Identifiers (DOIs) of the books in the corpus that can be used to reconstruct the original dataset (Kokash et al., 2023b).

The basic steps of the dataset processing pipeline are as follows:

1. Loop over the archive and search for JATS files.
2. Parse JATS files to retrieve names of PDF files containing bibliographic references and indexes.
3. Extract individual references and indexes from the corresponding PDF files.
4. Serialise publications, bibliographic references and index terms in the Neo4J database.
5. Retrieve any number of (unprocessed) bibliographic references from the DB and run HTTP requests to disambiguate them via Crossref and Google Book API services. Serialise results. Mark all processed references as "disambiguated".

6. Retrieve any number of (unprocessed) index references from the DB and run HTTP requests to disambiguate them via the Wikidata API. Serialise results. Mark all processed index references as "disambiguated".
7. Retrieve all references from DB, cluster them and save clusters in the database.
8. Retrieve all indexes from DB, cluster them and save clusters in the database.

Steps 5 and 6 can run in parallel, and each of these steps can be performed by several parallel processes. To avoid repeating disambiguation requests for the same data, it is useful to split references and index terms into non-overlapping sets. The processes can be interrupted at any time, e.g., if any of the processes fails because of network issues or unexpected exception in result parsing, it can be restarted again.

Steps 7 and 8 can be lengthy as we use an editing distance on pairs of bibliographic references or index terms to determine whether they address the same instance. We found it useful to save the UUIDs of created clusters and items they include in local files or locally deployed databases to avoid data loss in case the remote session gets interrupted.

For more details on the presentation and processing of reference and index files see Kokash et al. (2023a).

3. Knowledge Graph

– **The Brill Knowledge Graph deposited at Zenodo – doi:**www.doi.org/10.5281/zenodo.7691771
– **Temporal coverage:** 2006–2021

The pipeline parsed 1804 publications (the processing of 7 archives did not succeed because of the corrupted or unusually organised JATS files from which our parsing method was not able to extract the essential data). The extracted data is serialised in the form of the Neo4J graph with the schema shown in Figure 1. Each publication is represented in the KG by a node with the label "Publication", which keeps the extracted title, year of publication, publisher, and, if available, language, number of pages, and location. We also extracted and stored book identifiers (ISBN, DOI, etc.) in the nodes labelled "IndustryIdentifier", reachable from the publication via the relationship "HasIdentifier", and contributors (authors, editors, translators, etc.) in the nodes "Contributor" connected by the relationship "HasContributor".

The reference mining method extracted 372,636 references. They correspond to nodes with the label "Reference" in the KG and are linked to the publication by the relationship "Cites". The index mining algorithm extracted 572,399 index entries. They correspond to nodes with the label "IndexReference" in the KG and are connected to the publication by the relationship labelled "Includes". A bibliographic reference can be linked to an external record that disambiguates it via the relationship "RefersTo", and the external record, either a Google Books or Crossref URL, is represented by the node labelled "ExternalPublication". Similarly to the internal parsed publication, we link external publications to their contributors. An index term can also be linked to the external Wikidata reference dedicated to the subject via the "RefersTo" relationship, and the corresponding node is labelled "ExternalIndex".

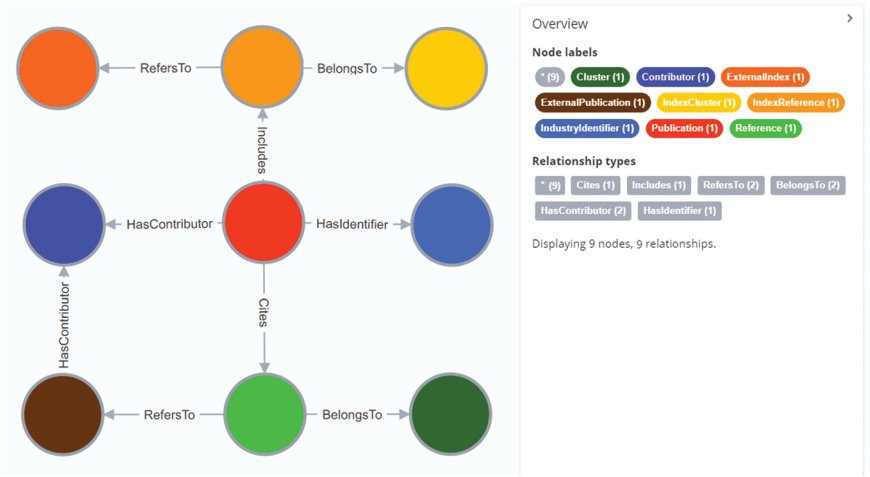

FIGURE 1 Resource types and relationships in the KG

For many references, we copied authors from the previous entries in the bibliography lists. Hence, in the KG the field "author" represents author as it is stated in the reference text while the "derived_author" represents either author from the reference text or from the preceding reference. After extracting the whole data, we discovered some references with omitted publication years. For such references, the publication year is also implied to be the same as in the preceding references, yet our initial implementation of the PDF bibliography parser did not account for such format.

The pipeline did not extract any bibliographic references for 803 books. The main reason for that is that we did not analyse the book content to locate bibliography spread across the chapters and/or mixed with text, which is quite common in edited volumes, as opposed to being provided in a separate file, which is usually the case in monographs. Besides, some manually inspected archives did not include bibliography data at all.

Table 1 lists titles of 5 books with the largest number of bibliographic references. It can be reproduced using Q1 from the Appendix.

TABLE 1 Book titles with the largest number of extracted bibliographic references

Book title	Number of references
"Brill's Companion to Ancient Greek Scholarship (Volume 1–2)"	3711
"Magnes"	3474
"Brill's Companion to Euripides"	3284
"La splendeur des dieux Quatre études iconographiques sur l'hellénisme égyptien"	3232
"Individuals and Materials in the Greco-Roman Cults of Isis"	2892

The average number of bibliographic references per publication is 206.56 (see Q2 in the Appendix).

For books with located bibliographic references, the information about the number of references per book as well as the number of references disambiguated via Google Books API (Google Books API S, 2012) and Crossref API (Crossref REST API, 2016) is given in Figure 2. Books are represented on the x-scale, ordered by the number of extracted references they contain, and the corresponding number of references per book is on the y-scale.

The total number of references disambiguated via the Google Books API is 97,054. This number can be obtained using Q3 in the Appendix. Note that the number of disambiguated references is somewhat smaller than the number

of nodes labelled "ExternalPublication" with type="google" (97,115). This is because one reference can be linked to several external resources representing the same published work: multiple matching links can be returned to a given request or there may be different volumes or versions of the cited work.

The total number of references disambiguated via the Crossref API is 69,745 (represented by 69,768 "ExternalPublication" nodes of type="crossref"). These numbers help us to establish that only **26%** and **18.7%** of references are disambiguated via Google Books API and Crossref API, respectively. In total, 144,474 references were disambiguated via two services which constitutes **38.77%** of all extracted references in the dataset (see Figure 4a). This is an important estimation that demonstrates that the most notable citation databases combined do not contain information about more than half of referred works. Admittedly, the disambiguation procedure contains errors (a human-based evaluation conducted in [Kokash et al., 2023a] estimates its precision to be 80.8%), therefore we cannot assume that these numbers are entirely correct, yet they provide a rough estimation for the citation database coverage in the field of AHSS. Parsing and reference extraction errors can also affect this estimation. For a more extensive evaluation and discussion of all pipeline steps, see the section *Data Quality* below.

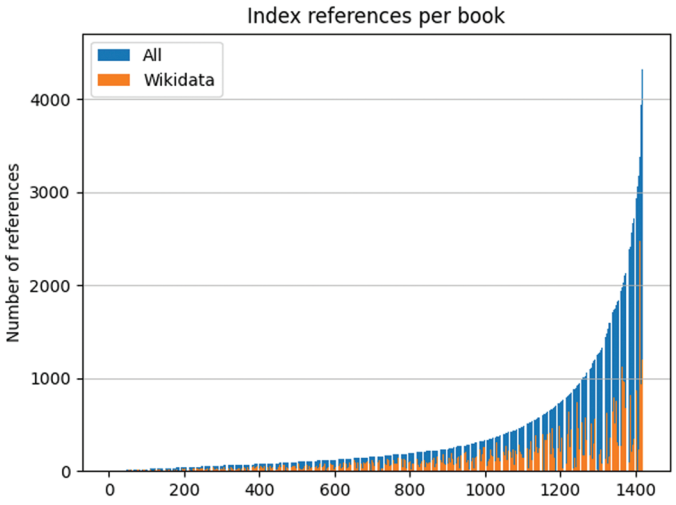

FIGURE 2 Number of bibliographic references per publication

The pipeline identified and parsed index files for 1421 books. Figure 3 shows the total number of indexes as well as the number of disambiguated via Wikidata indexes per publication. Table 2 lists titles of 5 books with the largest number of extracted indexes.

TABLE 2 Book titles with the largest number of extracted indexes

Book title	Number of indexes
"Individuals and Materials in the Greco-Roman Cults of Isis"	4483
"Plotinus on Love: An Introduction to His Metaphysics through the Concept of Eros"	4437
"Israel in Egypt"	4409
"History of Ancient Greek Scholarship"	4324
"Essen im antiken Judentum und Urchristentum"	4314

The average number of index terms per publication is 317.3 (see Q4 in the Appendix).

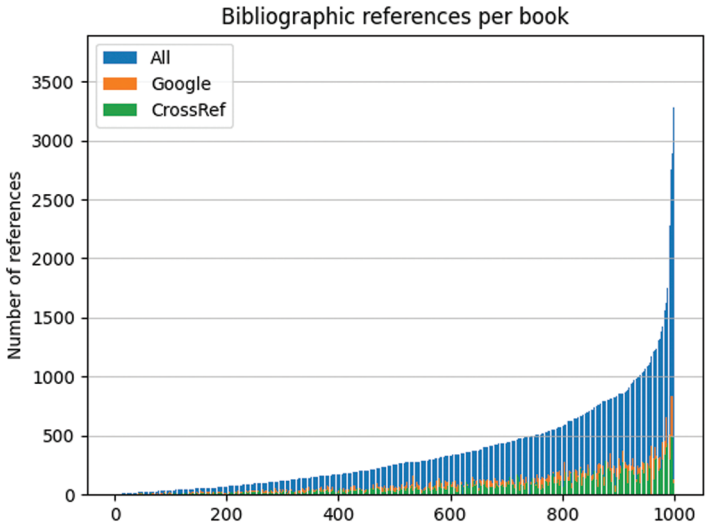

FIGURE 3 Number of indexes per publication

Note that in the KG, 21,986 index references are linked to more than one Wikidata article. This can happen because of:

- ambiguous labels, e.g., "Alexander" can refer to "Alexander The Great" (www.wikidata.org/wiki/Q8409) or to "Alexander VI", Pope of the Catholic Church (www.wikidata.org/wiki/Q108316);
- several Wikidata articles dedicated to various aspects related to the term, e.g., "Telemachus" is mentioned as a mythological personage (www.wikidata.org/wiki/Q192482), and a painting (www.wikidata.org/wiki/Q80081524);
- different representation of the same item in Wikidata, e.g., "Girolamo Cardano" (www.wikidata.org/wiki/Q63851404) vs "Cardano, Girolamo" (www.wikidata.org/wiki/Q103849876);
- composite or hierarchical indexes. For example, the index:

  Popes (Roman) 241
  - Alexander 69
  - Anacletus 204, 216
  - Anicetus 46
  - Callixtus 163
  - ...

  is linked to 48 Wikidata articles: "Anacletus" www.wikidata.org/wiki/Q80450, "Anicetus" www.wikidata.org/wiki/Q546590, "Callixtus" www.wikidata.org/wiki/Q122376, etc. This is a limitation of our pipeline that flattens multi-level index entries and represents them by a single "IndexReference" while it could be beneficial to serialise it as a hierarchy of related index terms.

To produce the bar plot of disambiguated indexes in Figure 3, we counted index terms per publication for which at least one external link was discovered (see Q5 in the Appendix).

In total, 191,546 out of 572,399 indexes, or **33.46%,** were disambiguated via Wikidata API (see Figure 4b), which can be estimated by the following query:

MATCH (*a:IndexReference*) *WHERE* (*a*)-[*:RefersTo*]-(*:ExternalIndex*) *RETURN count*(*a*).

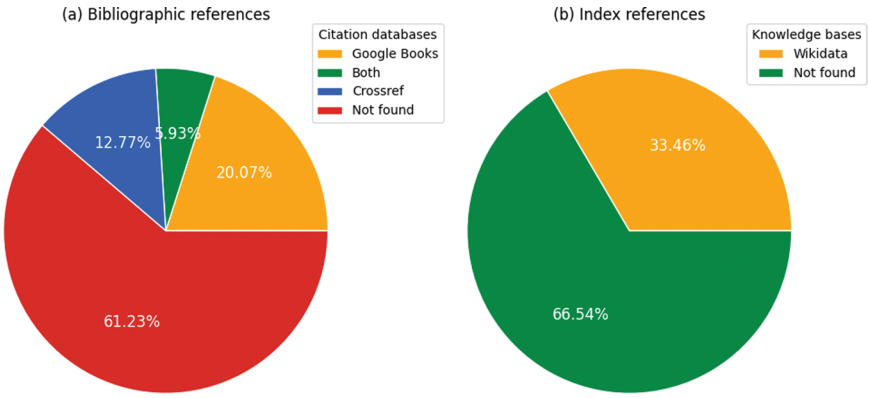

FIGURE 4A-B Fraction of disambiguated references

4. Clustering

Bibliographic references that connect to the external resource nodes with the same URL refer to the same published work. Hence, it is useful to combine them into clusters. Such clustering, however, is compromised by the pitfalls of the disambiguation process, i.e., absence of corresponding records in Google Books and Crossref citation databases. Similarly, extracted index terms may not be matched to any dedicated article in Wikidata. Yet, it is still useful to combine bibliographic references to the same work and index terms defining the same thing across the processed publication archive. We achieve this by computing editing similarity on pairs of parsed reference and index fragments, placing two bibliographic references to the same cluster if this ratio exceeds 75% and two index entries if their similarity is over 90% (see Kokash et al. [2023a] and Maarif et al. [2014] for the explanation on the choice of these thresholds). For the given dataset, 45,973 index clusters, which cover 197,351 index terms, and 50,730 bibliographic reference clusters, which cover 143,689 references, were created. A cluster node is created when at least 2 text references are found to be sufficiently similar. Hence, the books from the parsed corpus cite 279,677 unique published works and index 421,055 unique subjects.

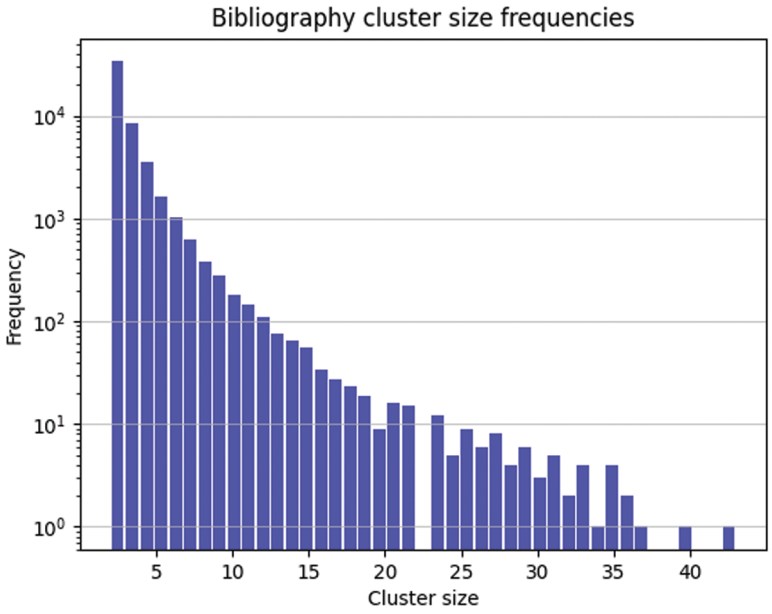

FIGURE 5 Number of bibliography reference clusters of a given size

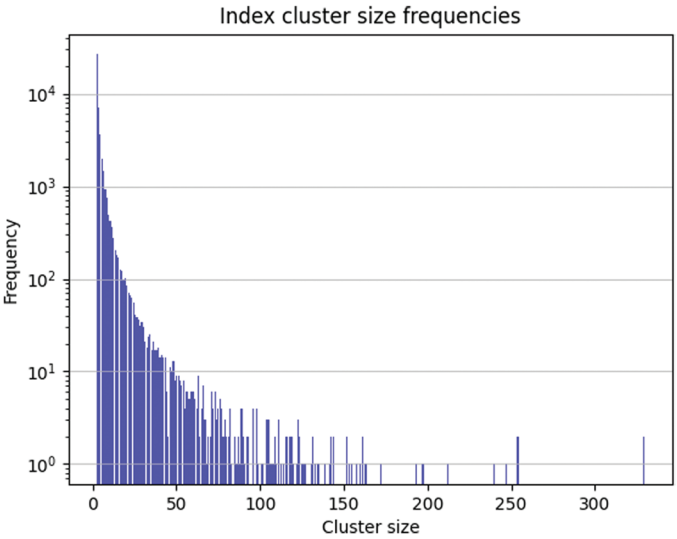

FIGURE 6 Number of index clusters of a given size

Figure 5 shows the distribution of bibliographic cluster sizes in the corpus. It may be interesting to look at k largest clusters of bibliographic references as they represent most cited works in a given collection of books (see Q6 in the Appendix).

Figure 7 shows the largest cluster which joins 43 extracted bibliographic references to the book *"Ptolemaic Alexandria"* by P.M. Fraser (http://dx.doi.org/10.2307/4348293, www.googleapis.com/books/v1/volumes/3z4KAQAAIAAJ).

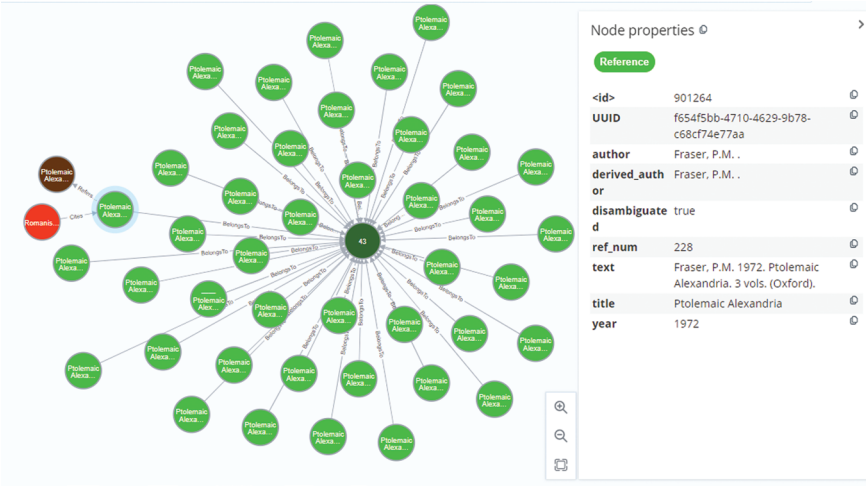

FIGURE 7 The largest cluster of bibliographic references

Table 3 provides more examples of bibliographic references corresponding to the most cited works discovered via clustering. In the first column, one or more sample text references are given. In the second column, we list the number of syntactically similar references used in the books from the Brill's dataset. Note that the method may link together references to apparently different books like this happened with our 2nd largest cluster in which most references contain "Bibliography" in their text.

The overall distribution of index cluster sizes in the corpus is shown in Figure 6.

Query Q7 from the Appendix reveals two clusters containing 330 index terms each referring to "Plato" and "Aristotle". Figure 8 shows the cluster of index references to "Plato". By extending the query to join the related publications, index clusters allow us to assemble collections of books dedicated to a certain topic.

TABLE 3 Samples of bibliographic references to the most cited works clustered together

Bibliographic reference samples	Occurrences
Fraser P. M., Ptolemaic Alexandria, 3 vols. (Oxford 1972).	43
Bibliography Rome, Madison, Wisconsin. Bibliography cols. CXXVII–CXL., Bibliography arabe. 2 vols. J. Gabalda. Paris., etc.	40
Bowersock, G.W. 1969. Greek Sophists in the Roman Empire. Oxford.	37
Cruz Andreotti, G. 20152g. s. v. 'Conistorgis.' In Cruz Andreotti, García Quintela, and Gómez Espelosín, eds. 396–97.	36
Dodds, E.R. (1951) The Greeks and the Irrational (Berkeley and Los Angeles)	36
"Horden, P. and Purcell, N. The Corrupting Sea: A Study of Mediterranean History (Oxford 2000)."	34

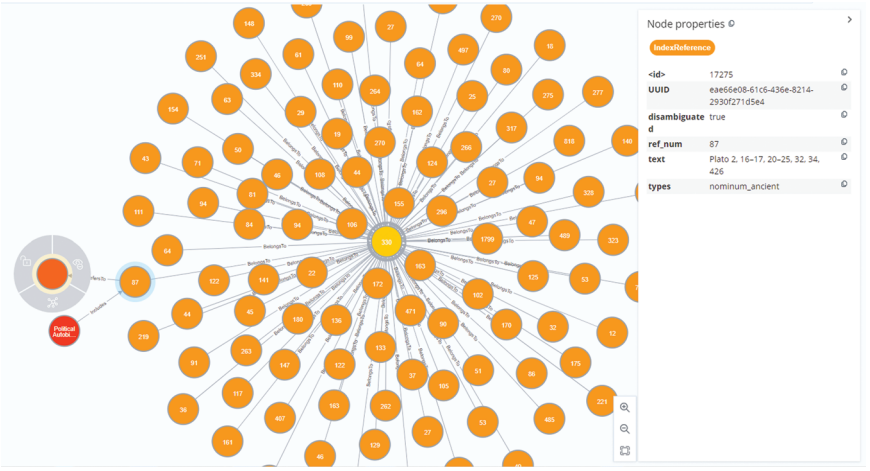

FIGURE 8 The largest cluster of index terms

Table 4 lists index mentions corresponding to the most cited authors discovered via clustering. In the first column, one or more index mentions are given. In the second column, we list the number of syntactically similar index terms used in the dataset.

Figure 9 shows the number of index terms per each type we assigned based on characteristic keywords used in the index file titles. Such classification is useful as specialised index lists are often created with certain format assumptions and conventions and more detailed parsers and structured data formats can be applied to store various index types.

TABLE 4 Samples of index mentions of the most cited authors clustered together

Index reference sample(s)	Occurrences
"Aristotle 24, 46, 99" "Aristotle 671, 673, 683f., 698, 729, 736, 738, 747, 806,810" "Aristotle, 140~1, 153~4, 156~7, 164, 188"	330
"Plato 4–5, 60, 64, 103, 163, 176, 197, 223, 312, 347"	330
"Plutarch, 111, 135, 141, 142, 170, 202, 226, 241, 521"	254
"col. 71.1671" "col. 16.1–15" "col. 42.26–3543"	254
"Homer 41, 124, 126, 139n, 186, 276, 278, 279, 280, 281, 297, 338, 385–6, 420, 516–19, 520, 526" "Homer56n64, 99, 101n28, 107, 148, 149, 164, 165, 259, 280" 281"	247
"Cicero 18n, 21"	240
"Cicero, 12, 246–264 passim, 343, 345, 349, 351, medicine, 42, 48, 50, 56"	

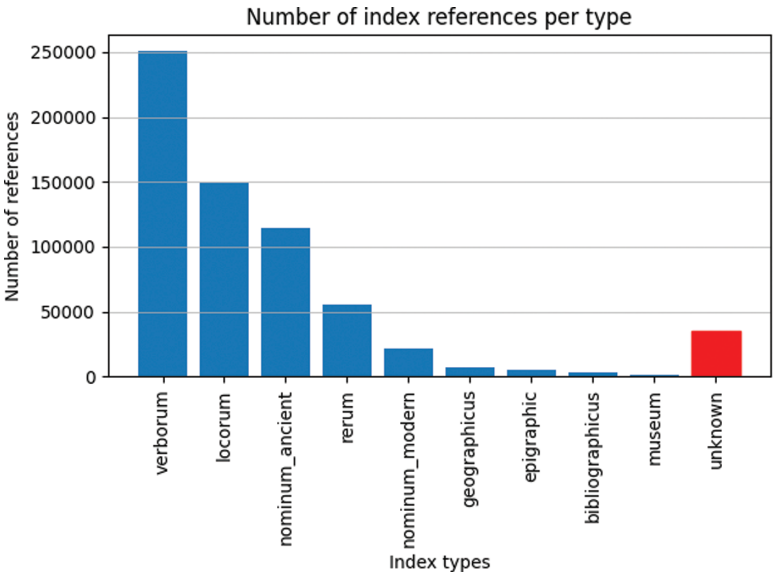

FIGURE 9 Classification of index terms

5. Data Quality

As the accompanying paper explains (Kokash et al., 2023a), because of the presence of bibliography files with mixed content (e.g., paragraphs of text in front or between lists of bibliographic references), it is possible that some other text got confused with bibliographic references. We do not know any clear syntactic rule that would allow us to reliably filter such entries. One of the heuristics to locate potentially mis-parsed references is "Reference" nodes without publication year. We see that only for 23,993 references our parser failed to extract the publication year, or 6.4% of the total number. It is important to note that the majority of these nodes represent valid references, and some were even disambiguated (5482). Table 5 shows such examples.

For our future work, we are considering training a machine learning classifier to distinguish correctly extracted references from garbage. Meanwhile, we can investigate the sanity of the extracted data by checking whether various queries yield meaningful and/or believable results. For example, a query to show publications referring to the works of Homer (see Q8 in tje Appendix) returns records like those shown in Table 6.

TABLE 5 Examples of valid bibliographic references without publication year

Bibliographic references without publication year
"Thomas Aquinas. In psalmos Davidis expositio Opera omnia, vol. 14. pp. 148– P. M. edita, vols. 4–5. Rome: Typographia Polyglotta S.C. de Propaganda Fide".
"———. forthcoming a. "The Pronomos Vase and Choregic Dedication." In Pronomos: His Vase and its World, ed. O. Taplin and R. Wyles. Oxford."
"———. forthcoming b. Icons and Actors: Six Studies in the Social and Economic History of Ancient Actors. Oxford."
"———. forthcoming. "Veiled Venom: Comedy, Censorship and Figuration." In Singing the Muses: Essays in Honor of Pietro Pucci, ed. C. Tsagalis and P. Mitsis. Berlin and New York."
"Vaticanus Ottobonianus gr. 188, 15th century, based on Vaticanus gr. 253 and 13th century, derived from Marcianus gr. 210."
"Vindobonensis phil. gr. 100 (J) 9th century (second half), De caelo on folios 56–214 with variants from Marcianus gr. 211."

TABLE 6 Examples of publications from the parsed archive citing "Homer"

Publication (title, year)	Index reference text
Theophrastus On First Principles (known as his Metaphysics) (2010)	"Homer 99n9, 120–121, 180–181, 235, 292"
Poetry as Window and Mirror (2011)	"Homer 10, 13, 24, 57–59, 61, 63– 69, 71, 72, 77, 78, 81, 82, 104, 105, 143, 168, 189, 209, 212–214, 229, 234, 235, 236, 238, 240, 247, 250, 251"
"Diodorus' Mythistory and the Pagan Mission" (2011)	"Homer 10, 11, 123n. 28, 125, 190, 256, 266, 313n. 22, 317, 332"

TABLE 6 Examples of publications from the parsed archive citing "Homer" (*cont.*)

Publication (title, year)	Index reference text
"Sacred Words: Orality, Literacy and Religion" (2011)	"Homer 17–34 (passim), 255–274 (passim)"
"Horace's Iambic Criticism" (2012)	"Homer, 16n32, 30n63, 54, 66, 71n66, 37, 43, 115n70, 121, 151, 231–"
"Cosmographia Christiana" (2014)	"Homer 25. 131. 171. 265. 378; Erfinder der Geographie 11"

More results can be found using the Wikidata identifier "www.wikidata.org/wiki/Q6691" as the author is known and referred to by different names in different languages, namely, ["Homer", "Homerus", "Homeros"]. The query Q8 that filters data on the text label collects 244 records, while the query that relies on the uri provided by the disambiguation services (see Q9 in the Appendix) collects 264 records. Both numbers are close to the 247 index mentions of Homer found by the clustering method.

6. Performance

The processing of the whole corpus to build the presented KG using the developed software was a rather lengthy process. The most time-consuming operation is reference disambiguation using external APIs. Some overhead was also added by DB serialisation operations as we were writing extracted data directly to a cloud-based Neo4j instance.

It took 1–2 seconds to execute a disambiguation request, create an external index reference node, and store it in the database. With just one process per each operation, that would require 6–10 days for all extracted index terms. The disambiguation of a bibliographic reference would require 5–7 seconds, or 21–25 days for all. However, we created 16 processes for bibliographic reference disambiguation and 16 processes for index term disambiguation, assigning to each of them entries with UUIDs starting from a given symbol (see Q10 in the Appendix). This simple parallelization strategy ensures that data subsets processed by each independent process do not overlap, and the entire dataset

was mined within 48 hours using an ordinary laptop (Lenovo Thinkbook 15 Gen 2 – i7-1165G7 – 16 GB – 512 GB SSD – Windows 11).

Each extracted "Reference" or "IndexReference" is processed independently and we create an "ExternalPublication" or an "ExternalIndex" node every time we find a match between the text of the extracted reference and disambiguation options provided by the disambiguation services. This may cause the presence of node "clones" in the KG: several nodes representing the same external resource. Similarly, contributors extracted from the structured data provided by Google Books and Crossref APIs are represented by separate nodes. Such nodes can easily be unified with the help of the APOC (Awesome Procedures on Cypher, 2016) library that provides a useful apoc.refactor.mergeNodes method. However, we decided not to merge node clones because automated disambiguation is not completely reliable. Faulty associations can be made between references and external resources, e.g., an index mentioning Pope Alexander VI gets linked to Wikidata's article about Alexander the Great: if such an erroneous relationship is detected by a human reviewer, it is easier to correct it by revising just one relationship rather than consider how the change affects other reference nodes pointing to the same node. For the same reason, we do not merge "Contributor" nodes.

7. Conclusions

We presented a KG and examples of queries that provide statistical data or interesting facts about bibliographic references and indexed terms in AHSS books shared with us by their publisher. A significant potential of the pipeline we developed lies in its applicability to parse other scholarly PDF archives (with perhaps minor modifications to accommodate organisational rules of large archives adopted by their owners, i.e., folder structure, naming conventions, annotation format, etc.), thus contributing to a faster population of citation databases with missing information.

We see three further uses. First, our dataset makes it possible to add authority information for the books cited in this corpus to OpenCitations or Wikidata. The fact that only 38.77% of extracted references could be matched against Google Books or CrossRef shows a major issue with the coverage of the classics domain in these citation databases. To mitigate this issue, one could revise the clusters of references, link authors and various bits of information to Wikidata entries, and make this a valuable resource for citation matching in the field of Classics.

Second, our work allows the creation of citation networks, including data from both secondary (bibliographies) and primary literature (indexes). With some further processing of the lists of pages where a certain index entry occurs, it would be possible to (re-)construct such networks. They could be used for bibliometrics analysis (like Blidstein and Zhitomirsky-Geffet, 2022) as well as to power bibliographic recommendation systems.

Third, our dataset can be used to train classification models to distinguish bibliographic references from garbage, and, similarly, index entries from other data that could have been erroneously considered to be an index: textual representation of disambiguated references and index terms can serve as examples of valid entries, while random sentences or their parts as "garbage".

All its caveats notwithstanding, we are confident that our work can contribute to research on Classics by analysing the network of references in the Brill Classics corpus, and can serve as an example of a scholarly knowledge base in the Humanities, that other publishers could consider implementing and releasing.

Acknowledgements

This research was supported by an NWO grant (dossier number KIEM.K20.01.137) and, in part, by the University of Amsterdam Data Science Centre.

References

Awesome Procedures On Cypher (APOC) (2016). *Documentation* [Software]. www.neo4j.com/labs/apoc.

Blidstein, M., & Zhitomirsky-Geffet, M. (2022). Towards a new generic framework for citation network generation and analysis in the humanities. *Scientometrics*, *127*(7), 4275–4297. doi: 10.1007/s11192-022-04438-y.

Crossref REST API (2016). *Documentation* [Software]. www.crossref.org/documentation/retrieve-metadata/rest-api.

Google Books APIs (2012). *Volume Books API v1* [Software]. https://developers.google.com/books/docs/v1/reference/volumes.

Kokash, N., Romanello, M., Suyver, E., & Colavizza, G. (2023a). From books to knowledge graphs. *Journal of Data Mining and Digital Humanities*, 2023 (March 13, 2023), doi: 10.46298/jdmdh.9380.

Kokash, N., Romanello, M., Suyver, E., & Colavizza, G. (2023b). The Brill Knowledge Graph: A database of bibliographic references and index terms extracted from books in humanities and social sciences. Open Access Dataset [Data set]. doi: 10.5281/zenodo.7691771.

Linder, S. K., Kamath, G. R., Pratt, G. F., Saraykar, S. S., & Volk R. J. (2015). Citation searches are more sensitive than keyword searches to identify studies using specific measurement instruments. *Journal of Clinical Epidemiology*, *68*(4), 412–417. doi: 10.1016/j.jclinepi.2014.10.008.

Maarif, H. A., Akmeliawati, R., Htike, Z. Z., & Gunawan, T. S. (2014). Complexity Algorithm Analysis for Edit Distance. 2014 International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia (pp. 135–137). doi: 10.1109/ICCCE.2014.48.

Naik, G., & Pai, R. (2020). Role of citation databases in research: A study. *Library Philosophy and Practice* (*e-journal*).

Nazarovets. M. (2021). Citation databases: the role of the academic librarian. University Library at a New Stage of Social Communications Development. Conference Proceedings, *6*, 118–123. doi: 10.15802/unilib/2021_248693.

Prasetya, D., Wibawa, A., & Hirashima, T. (2018). The performance of text similarity algorithms. *International Journal of Advances in Intelligent Informatics*, *4*. doi: 10.26555/ijain.v4i1.152.

Rahm, E., & Thor, A. (2005). Citation analysis of database publications. *ACM SIGMOD Record*. *34*(4), 48–53. doi: 10.1145/1107499.1107505.

Appendix

Cypher queries to reproduce statistical results in this publication:

Q1 `MATCH (p:Publication)-[c:Cites]-(r:Reference) RETURN p.title, count(r) as c order by c desc limit k`

Q2 `MATCH (p:Publication) WITH p, size((p)-[:Cites-()) as refCount RETURN avg(refCount).`

Q3 `MATCH (a:Reference) WHERE (a)-[:RefersTo]-(:ExternalPublication {type:"google"}) RETURN count(a)`

Q4 `MATCH (p:Publication) WITH p, size((p)-[:Includes]-()) as refCount RETURN avg(refCount).`

Q5 `MATCH (p:Publication)-[:Includes]-(r:IndexReference) WHERE size((r)-[:RefersTo]-(:ExternalIndex)) > 0 RETURN p.UUID, count(r).`

Q6 `MATCH (m:Cluster) RETURN m order by m.size desc limit k`

Q7 `MATCH (m:IndexCluster) RETURN m order by m.size desc limit k`

Q8 *MATCH* (*a:* *ExternalIndex*)-[]-(*b:IndexReference*)-[]-(*c:Publication*) *WHERE* *a.text*=*"Homer"* *RETURN* *c.title, c.year, b.text*

Q9 *MATCH* (*a:* *ExternalIndex*)-[]-(*b:IndexReference*)-[]-(*c:Publication*) *WHERE* *a.uri*=*"www.wikidata.org/wiki/Q6691"* *RETURN* *count*(*a*), *collect* (*distinct* *a.text*).

Q10 *MATCH* (*a:Reference*) *WHERE* *a.disambiguated is NULL* *RETURN* *substring*(*a.UUID, 0,1*) *as* *first, count*(*a*) *order by* *first*