Linked Open Citation Database: Enabling Libraries to Contribute to an Open and Interconnected Citation Graph

Anne Lauscher
University of Mannheim
Mannheim, Germany
anne@informatik.uni-mannheim.de

Kai Eckert
Stuttgart Media University
Stuttgart, Germany
eckert@hdm-stuttgart.de

Lukas Galke
ZBW – Leibniz Information Centre for Economics
Kiel, Germany
L.Galke@zbw.eu

Ansgar Scherp
ZBW – Leibniz Information Centre for Economics
Kiel, Germany
A.Scherp@zbw.eu

Syed Tahseen Raza Rizvi
DFKI – German Research Center for Artificial Intelligence
Kaiserslautern, Germany
syed_tahseen_raza.rizvi@dfki.de

Sheraz Ahmed
DFKI – German Research Center for Artificial Intelligence
Kaiserslautern, Germany
sheraz.ahmed@dfki.de

Andreas Dengel
DFKI – German Research Center for Artificial Intelligence
Kaiserslautern, Germany
andreas.dengel@dfki.de

Philipp Zumstein
Mannheim University Library
Mannheim, Germany
philipp.zumstein@bib.uni-mannheim.de

Annette Klein
Mannheim University Library
Mannheim, Germany
annette.klein@bib.uni-mannheim.de

ABSTRACT

Citations play a crucial role in the scientific discourse, in information retrieval, and in bibliometrics. Many initiatives are currently promoting the idea of having free and open citation data. Creation of citation data, however, is not part of the cataloging workflow in libraries nowadays.

In this paper, we present our project Linked Open Citation Database, in which we design distributed processes and a system infrastructure based on linked data technology. The goal is to show that efficiently cataloging citations in libraries using a semi-automatic approach is possible. We specifically describe the current state of the workflow and its implementation. We show that we could significantly improve the automatic reference extraction that is crucial for the subsequent data curation. We further give insights on the curation and linking process and provide evaluation results that not only direct the further development of the project, but also allow us to discuss its overall feasibility.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**; *RESTful web services*; *Resource Description Framework (RDF)*; *Search interfaces*; *Link and co-citation analysis*; • **Applied computing →** *Optical character recognition*;

JCDL '18, June 3–7, 2018, Fort Worth, TX, USA

ACM ISBN 978-1-4503-5178-2/18/06…$15.00
https://doi.org/10.1145/3197026.3197050

KEYWORDS

citation data; library workflows; linked open data; editorial system; automatic reference extraction

ACM Reference Format:
Anne Lauscher, Kai Eckert, Lukas Galke, Ansgar Scherp, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, Philipp Zumstein, and Annette Klein. 2018. Linked Open Citation Database: Enabling Libraries to Contribute to an Open and Interconnected Citation Graph. In *JCDL '18: The 18th ACM/IEEE Joint Conference on Digital Libraries, June 3–7, 2018, Fort Worth, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3197026.3197050

1 INTRODUCTION

Originally, libraries cataloged their holdings only on the level of physical objects, mostly books, with the primary purpose to identify them and locate them on the shelves. Resources like conference proceedings, collections or journals are in the catalogs, too, but only on the level of a book or volume – the physical, bound book that is put somewhere on a shelf. The single papers in a conference proceeding, collection or journal, in which the patrons are usually most interested, are not captured in traditional library catalogs. This has changed since many libraries have introduced Resource Discovery Systems (RDS) that combine the classical content of the library catalog with large indexes of article data. These indexes are dominantly in the hand of commercial providers [2], but there are also library based approaches like the K10plus-Zentral index[1] that is curated by a library service center and at least partly contains article data produced by libraries. While libraries start to catch up on the indexing of articles, the database providers often already include another data source: citations extracted from the lists of references in the articles. Although such citation data gives a great functionality for a single database and allows to examine the referenced article as well as the articles referencing the current one, the

[1] https://verbundwiki.gbv.de/display/VZG/K10plus-Zentral

same data usually cannot be used in another database. The information which article cites a given article is extremely helpful during literature research [22] and allows a versatile ranking, not only of articles, but also authors, journals, conferences, and institutions, depending on the level of aggregation. [13]

Many libraries spend a lot of money for the big commercial citation databases like Web of Science (WoS) or Scopus. These databases have a focus on journal articles, although they have started some years ago to include other forms of scientific output as well, e.g. Book Citation Index, Data Citation Index. A comparative study from 2016 showed that the journal coverage of WoS and similarly for Scopus in Social Sciences is quite low and "English-language journals are favored to the detriment of other languages" [14]. It is not possible for libraries to extend the coverage of these databases by cataloging some missing resources. Moreover, there are restrictions about what you are allowed to do with the data, which exclude sharing the data in larger chunks with other researchers. This, however, is exactly what we would expect today from an open science perspective. For a transparent and replicable bibliometric or scientometric analysis we need open citation data.

Our project Linked Open Citation Database (LOC-DB)[2] aims to prove that a distributed infrastructure for the cataloging of citations in libraries is possible. We want to create a citation database within libraries, where the resulting citation data is available as linked and open data (under a CC0 waiver) using the OpenCitations data model to foster collaboration and sharing. Therefore, we develop a workflow (Section 3) that is embedded into the actual cataloging workflow of a library. The technical architecture (Section 4) is designed to distribute the workload across many participating libraries. At the core is the database back end (Section 4.1) that integrates components like the automatic reference extractor (Section 4.2) and various data sources for link suggestion. All data and links are curated in the front end editorial system (Section 4.3). All these components including the workflow need to be optimized to make the curation of the citation data feasible. We are particularly interested in the time that is needed to curate the citations and the quality of the resulting data that can be achieved. In Section 5, we present results of our initial evaluation after the first prototype implementation. This enables us to discuss some first estimations (Section 6) regarding our main question: how much would it cost, with respect to resources, if libraries cataloged everything and curated the citation graph?

2 RELATED WORK

Citation indexes in science first have been introduced by Garfield in 1955 [7] which lead to the development of the commercial database Web of Science owned now by Clarivate Analytics. Another big company with large influence in bibliometric analyses is Google which runs Google Scholar.[3] While Google Scholar is open for everyone to search and examine the citations in there, the data is not openly shared and cannot be used in other contexts or for large bibliometric analyses. Competing search engine providers have similar products, as Microsoft Academic[4] and Baidu Scholar.[5]

Other citation projects originate from academic institutions, like CiteSeer, that started in 1998 and since then was renamed to CiteSeerX,[6] hosted today by the Pennsylvania State University. They use a web crawler and a fully automated workflow for processing publicly available full-text documents in order to extract the citations [8], which sometimes leads to low quality citation data [9]. Their data can be reused under a CC-BY-NC-SA license.[7] The software is reused in CitEC - Citations in Economics[8] which analyzes full-text documents in economics and provides their data to the RePEc services. A similar fully automated pipeline based on the full-text of publicly available publications is run for Semantic Scholar[9] by the Allen Institute for Artificial Intelligence since 2015.

The OpenCitations project[10] has also a fully automated workflow but based on structured data, mainly from Europe PubMed Central. The references are then processed and provided as RDF-formatted data openly with a SPARQL end point. [16, 18] LOC-DB has adopted the OpenCitations data model, and is in close contact with the directors of OpenCitations to explore the possibilities for further cooperation.[11]

A mixture of automatic additions and manual edits is possible in the collaborative knowledge base Wikidata[12], which includes publication data and citation data mainly because of the recent WikiCite[13] initiative, with a focus on the automated extraction from electronic publications. [20]

OpenCitations and WikiCite are also among the funding organizations of the Initative for Open Citations (I4OC),[14] with the goal to promote the unrestricted availability of scholarly citation data. Several publishers already joined this initiative and provide their reference data openly via Crossref.[15] ExLibris includes this data into its commercial database Primo Central, that drives the Primo Discovery Service, since May 2016.[16] And of course we also use it in our workflow wherever possible.

From all these projects and products it can be seen that citation data is highly important and created by various means and from various sources, in varying quality. We believe that libraries should play a main role in providing this data for the public, and that fully automatic methods are not reliable enough considering the importance of the data. This is especially true for older publications which are not available in an electronic format. Hence, we reuse existing data and combine state-of-the-art reference extraction

[2] https://locdb.bib.uni-mannheim.de/
[3] https://scholar.google.com/
[4] https://academic.microsoft.com/, with a paid API https://azure.microsoft.com/en-us/services/cognitive-services/academic-knowledge/.
[5] http://xueshu.baidu.com/
[6] http://citeseerx.ist.psu.edu/
[7] http://csxstatic.ist.psu.edu/about/data
[8] http://citec.repec.org/
[9] https://www.semanticscholar.org/. They provide their data for download, but restrict the reuse on non-commercial research and non-commercial educational purposes: http://labs.semanticscholar.org/corpus/corpus/legal.
[10] http://opencitations.net
[11] https://opencitations.wordpress.com/2018/03/23/early-adopters-of-the-opencitations-data-model/
[12] https://www.wikidata.org/
[13] https://meta.wikimedia.org/wiki/WikiCite
[14] https://i4oc.org/
[15] https://www.crossref.org
[16] https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Highlights/047Primo_May_2016_Highlights, section Citation Trails

techniques with the knowledge of domain-experts, in an efficient system with distributed workflows.

3 LIBRARY WORFKLOW AND DATA

The capturing of citations in libraries can be integrated in the standard workflow for new acquisitions, or performed systematically in projects processing specific collections that might be of special interest. All possible publication formats, i.e. print and electronic, must be covered. For the LOC-DB project, a part of the social sciences collection of Mannheim University Library has been chosen to implement the workflow for the retrospective cataloging of citations. It consists of 522 print books and collections acquired by the library in 2011, and the articles that were published during the same year in 101 (mostly electronic) journals.[17] Apart from that, the new print acquisitions of the social sciences branch library are also processed regularly from July 2017 on.

The LOC-DB workflow builds on the traditional library cataloging work: Books, collections and journals already come with high quality metadata that can be retrieved in MARC21 format from a library union catalog. For collections and journals, the metadata of chapters and articles contained in them have to be captured next, in order to precisely identify the citing resources. All metadata of citing resources are saved to the LOC-DB database in a standardized, structured format that is fit for sharing and reuse in Linked Open Data contexts (see Section 4.1.2). Then, the references themselves can be ingested. If the citing resource is in print, the reference section is scanned and processed by the Automatic Reference Extractor of the LOC-DB system (see Section 4.2). If it is electronic, either the reference data can be retrieved from external sources (see Section 4.1.3) or the electronic text of the reference section is processed by the Automatic Reference Extractor. In any case, the output is a list of separate references (the "bibliographic entries"). They can be in a more or less structured or completely unstructured form, so they have to be "resolved", i. e., linked to a record in the standard LOC-DB format (a "bibliographic resource"). If possible, this is done by retrieving the metadata for the item from appropriate external databases (see Section 4.1.3). We take care to store the identifiers of the source databases (for example Crossref DOIs or union catalog IDs for books) in order to foster a possible future integration of the LOC-DB data into these databases.

3.1 Print Books

The set of 522 books consists of 330 monographs and 192 collections, for which a student assistant scanned the reference lists – usually one larger list at the end of a monograph or several shorter ones after each chapter of a collection. The average extent of a reference list in our set was 26 pages (see Table 1).

A standard book scanner has been used to create the scans. Comparable scanners are already available for patrons in many libraries, so there would be no need to purchase new hardware to do the workflow in many places.

For the new acquisitions, the relevant books are already marked with a special note in the acquisition process. When the books arrive, they pass through the normal cataloging workflow and other work steps of the library. For the collections, the librarians copy the table of contents, which will become handy later in the process. The books are then placed in a special location in our offices for the scanning process. If the item is not immediately requested by a patron, the scanning is done when a student assistant is available - this can take one or two days. Urgently needed books are of course provided to the patrons first, and the scanning is done later.

Table 1: Printed books in social sciences purchased 2011 by Mannheim University Library with the number of pages in the list of references.

	monographs	collections	total
no. of books	330	192	522
pages overall	6,690	6,860	13,550
pages avg.	20	36	26

3.2 Electronic Journals

Most journals provide some alerting services like an RSS feed for newly published articles or issues. For the remaining journals, the library can actively monitor them on a regular time base. The bibliographic metadata for most journal articles itself can be found in Crossref, although we do some additional work trying to connect to some library databases as well. The citation data on the other hand can be in a lot of forms, which all have to be considered:

- citation data is already available in LOD, e.g. by OpenCitations project
- citation data is available as open data (structured or unstructured), e.g. by Crossref
- publisher provides some API for structured article data in XML, e.g. JATS-XML or Springer's A++ format
- full-text including the references is available as HTML
- full-text including the references is only available as PDF

We so far focused on the second case which is already quite frequent, thanks to the Initiative for Open Citations. A first assessment of the quality of the relevant data in Crossref showed that although there is a considerable amount of public, structured references available, many of these references are missing important data fields (like DOI, title, and first-page), which makes the identification of these publications really hard.[18] Thus, in order to provide high-quality data this then still needs further processing, which we do in our editorial system.

4 LOC-DB ARCHITECTURE

Our workflows are supported by three infrastructure components which we develop in our project: The central LOC-DB component, i.e. the back end (Section 4.1), the automatic reference extraction (Section 4.2), and the editorial system, i.e., the graphical user interface exposed to the librarians (Section 4.3). An overview of

[17] An electronic version of all these journals exists, but in two cases, Mannheim University Library doesn't have access to it, so that the print version has to be used for the project.

[18] E.g. among the relevant references for LOC-DB in Crossref only 35% have a DOI and only 43% have a year. For the detailed analysis, see the Data Quality section in https://github.com/locdb/locdb-journal-analysis/blob/master/locdb-journals.ipynb.

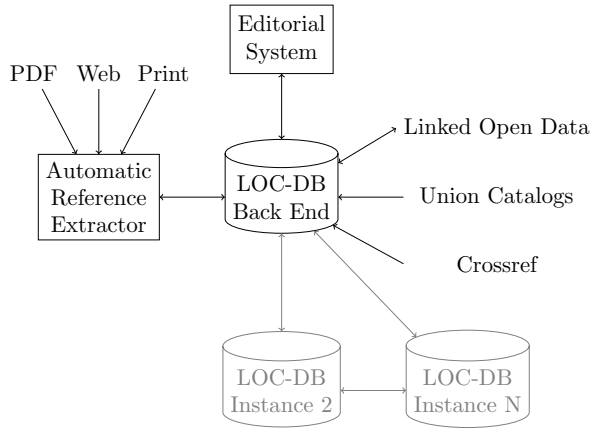


Figure 1: The infrastructure components of the LOC-DB project.

this infrastructure can be seen in Figure 1. The whole system is designed to work distributed, with independent instances of LOC-DB, communicating with each other to provide lookup services.[19]

4.1 Database and Back End

The back end of our system acts as a distributed central hub, which is responsible for saving files and data, ensuring the communication between all three components as well as for retrieving metadata from a variety of external sources. In the following, we will first describe the general architecture, then the provisional data model and last how we derive metadata from external sources.

4.1.1 Layered Design. The central LOC-DB component has two layers, the storage layer and the web-based Application Programming Interface (API). As data base in the storage layer we use MongoDB,[20] which we combine with an Elasticsearch[21] index in order to efficiently provide internal suggestions for matching bibliographic resources for a given citation to the user. Furthermore, we save files, such as uploaded scans of reference pages, in the file system. On top of that we expose several HTTP end points to the front end, which we implemented using Node.js.[22] These end points provide user management, basic `Create`, `Read`, `Update`, `Delete` (CRUD) functionalities as well as triggers for more sophisticated services such as the automatic reference extraction or the retrieval of external suggestions. A Swagger[23] interface which serves as interactive documentation describing the current status of the HTTP end points[24] as well as our source code[25] is available online.

4.1.2 Data Model. In order to ensure the sustainability of the data we generate and a seamless integration in the linked open data cloud, we adapt the OpenCitations metadata model [17] for our purposes. The model is aligned with the Semantic Publishing and Referencing (SPAR) Ontologies [15] for creating machine-readable metadata describing citations in Resource Description Framework (RDF) format. The main bibliographic entities of the OpenCitations data model are the following:

- Bibliographic Resource: A bibliographic resource, e.g. a monograph, which is cited by or cites another bibliographic resource.
- Resource Embodiment: The embodiment of a bibliographic resource, which can be print or digital.
- Bibliographic Entry: An entry in a reference list of a bibliographic resource, i.e. a citation.
- Responsible Agent: An agent, who is responsible for a bibliographic resource, e.g. a person.
- Agent Role: The role of an agent, e.g. author or publisher.

We extend the data model mainly in two parts: First of all, a property `scan` is added to resource embodiments, which saves links from an embodiment of a bibliographic resource to files that hold references belonging to that resource. This way, given a bibliographic resource, one can retrieve the original references pages of that resource at any time. Secondly, we add a link from each bibliographic entry to the scan it appears on, if available. The two extensions of the model ensure that we can not only automatically extract references from PDF files or image scans, but also that we are able to present the original page to the librarians when they check the data and create the links.

4.1.3 External Metadata Retrieval. Acting as a central hub, we implement back end services to extract metadata from a variety of external sources. Ideally, we get high-quality structured data this way which clearly speeds up the curation process. Another aspect is the creation of links which is automatically done whenever the librarian accepts a match. For the creation of bibliographic resources, we build upon the already existing cataloging infrastructure. For example, when a print monograph arrives in the library, it has been already assigned a union catalog identifier and also the metadata for that item has been already curated in the library union catalog. We reuse the cataloged metadata and retrieve it using existing end points, e.g. Search/Retrieve via URL (SRU)-interfaces, convert it, e.g., from MARC21, into our data model, and keep the original identifier to link back to the catalog.

For the cited bibliographic resources, where metadata might not yet be in our database, the user receives matching external suggestions, as stated above. The user ideally just has to select the suggestions that fit in order to create metadata for the target bibliographic resource, where links are created back to all matching resources. We encourage users to select all matching resources to increase link coverage. There is, however, also an immediate benefit when more resources are selected, as the metadata is created by combining all matched resources. This way, missing information can be filled and in case of differences in the data, ranking approaches can be applied to improve the data quality.

External sources for metadata include Crossref, the OpenCitations SPARQL endpoint, Google Scholar, or more domain-specific bibliographic databases. Candidates are generated using a query

[19] The software components we develop in our project are going to be available with an OpenCitations approved open source license.
[20] https://www.mongodb.com/
[21] https://www.elastic.co/de/products/elasticsearch
[22] https://nodejs.org/en/
[23] https://swagger.io
[24] https://locdb.bib.uni-mannheim.de/demo/docs/
[25] https://github.com/locdb/loc-db

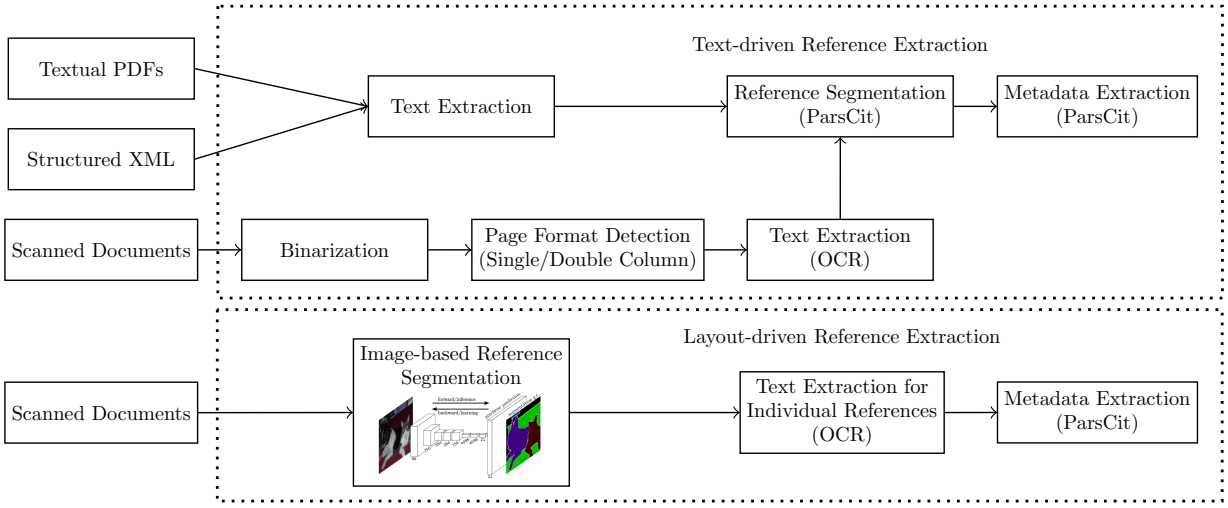


Figure 2: Automatic Reference Extraction Pipeline.

string based on the data we already have. In a second step, the candidates are weighted using the Dice similarity coefficient (DSC) [6]. The user can specify a threshold above which he or she wants to see the results or use the current default value of $DSC = 0.45$, which we observed to be a useful starting point in our user studies. We plan to further adapt this default value in order to provide optimal linking assistance and to conduct experiments with other measures of string similarity, e.g. Levensthein distance [11].

With this procedure, we want to ensure that the user does neither need to search manually for matching metadata in multiple metadata provider catalogs nor to enter all the metadata manually.

4.2 Automatic Reference Extraction

For the extraction of references, we use an automated approach that processes the pages with references, which might be scanned or born-digital.

Figure 2 provides an overview of the automatic reference extraction pipeline, which consists of two sub components: Text-driven Reference Extraction and layout-driven reference extraction.

4.2.1 Text-driven Reference Extraction. This method uses textual information to detect bibliographic references from a given document and is highly dependent on the referencing style of the document. Text-based extraction can be used for different electronic resources (born digital PDFs, structured xml, etc.) and scanned documents. For electronic resources, the document is passed directly to a text extraction module followed by reference segmentation and metadata extraction, i.e. to classify elements like author, title, journal, etc. For reference segmentation and metadata extraction, ParsCit [5] is used. ParsCit is an open source package for detecting and labeling citations strings from text. It is based on Conditional Random Fields (CRF) along with sophisticated heuristics to detect and segment citations strings.

For processing a scanned document image, it is firstly preprocessed by performing binarization (converting an RGB image to a binary image). The preprocessed document is further classified into a single column or double column layout document during page format detection phase. Once classified, the image is then passed to an optical character recognition (OCR) process for text extraction. We use OCRopus [3] which is an open source OCR software using Long Short-Term Memory (LSTM) networks [4]. Similar to electronic resources, the text obtained from OCR is then passed to the reference segmentation module to identify the reference and extract metadata out of the detected string. Both segmentation as well as metadata extraction is performed using ParsCit.

Text-driven extraction alone has several drawbacks. First, it depends heavily on the quality of OCR. If the text is not correctly recognized by OCR, then whole references can be missed. Second, for detecting and segmenting references, this approach relies on ParsCit. While ParsCit can be seen as state-of-the-art, it sometimes is unable to detect some references or it merges multiple references together and classifies them as a single reference. We therefore seek to improve the process by complementing the text-driven extraction with layout-driven extraction.

4.2.2 Layout-driven Reference Extraction. In contrast to the text-driven extraction, the layout-driven reference extraction uses only the layout information to localize each reference string in the document image. The method is inspired by how humans interpret a document. In this approach, the goal is to use layout information to localize each reference string region and then that region is passed to OCR for text extraction. To do so, a deep learning-based approach [1] is used to train a model for segmentation of individual references based on the graphical layout. Therefore, during training and application, all the text lines are blurred so that the content is no more readable. The trained model is then used to segment reference regions from new unseen scanned images. OCR is performed on detected reference areas of the image. Once the OCR is completed, all the text of the reference string is passed to the ParCit for metadata extraction.

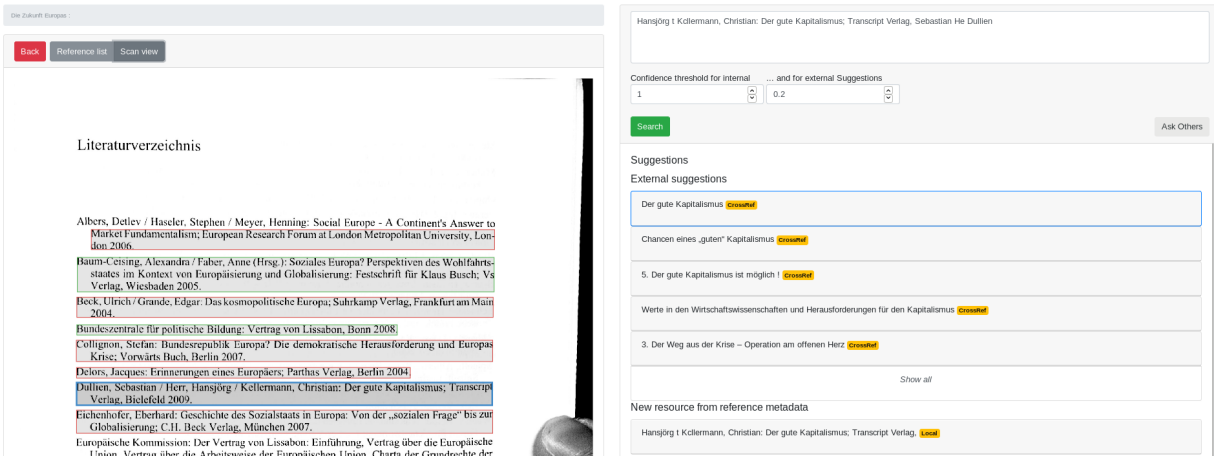


Figure 3: Reference linking in the editorial system. On the left: the reference list of the citing document. On the right: metadata of the candidate target resource.

4.3 Editorial System

The main goal of the editorial system is to provide a graphical user interface for librarians, wrapping the functionality of the LOC-DB back end as well as the automatic reference extractor. We developed the editorial system according to the principles of Interaction Design [19]. We created wireframes to mock the key words and visual representations of the interaction elements while taking the typical working environment of librarians into account. After discussing these wireframes with librarians from both ZBW and University Library of Mannheim, we implemented a prototype for continuous testing. This way, we could already identify and resolve early usability issues and proceed to continuously enhance the editorial system in accordance with the librarians.

4.3.1 Ingestion. The editorial system is split into two major visual components. On the one hand, there is a data ingestion component which allows uploading electronic documents, scanned documents, as well as providing an identifier to ingest metadata from external sources. When any kind of data is ingested, the resource metadata is retrieved (see Section 4.1.3) and the automatic reference extraction is triggered (see Section 4.2). On the other hand, the reference linking component offers the functionality to inspect the reference list of the source document and assign a matching target bibliographic resource for each of the citations.

4.3.2 Reference Linking. Resolving a citation link is a complex process that requires viewing the reference list of the source document along with the metadata of candidate target documents. We tackle this challenge by splitting the user interface into two columns (see Figure 3). On the left, the reference list of the source document is presented. On the right, the metadata of possible citation targets is displayed. To ensure that resolving a scientific reference can be performed in reasonable time, the editorial system presents the suggestions for matching resources generated by the back end (Section 4.1.3). The suggestions comprise data from the internal database as well as external data sources. In this context, internal means that a matching record has already been inserted into the LOC-DB earlier. External refers to those suggestions that come from external data sources. When only external suggestions are selected as a citation target, the metadata will get copied to the internal database, such that it can be edited if necessary. Hence, the formerly external resource will then be available as an internal suggestion. To avoid duplicate records, identifying a matching internal record is crucial and has the highest priority. In the worst case, neither internal nor external suggestions offer a match to the currently processed entry of the reference list. In this case, the LOC-DB system is still able to use the structured data which was extracted from the reference in the source document as a starting point for a new resource record. The front end needs to reflect this process of first looking for internal duplicates, then finding as many external references as possible and finally, as a last resort, providing the means to create a new record from scratch.

The editorial system needs to be capable of dealing with scanned documents, electronic documents, as well as metadata from external sources. An appropriate display of the reference lists is required for each of the three cases. For scanned documents and electronic documents, the reference list is displayed as an image with bounding boxes indicating automatically identified references. On the other side, sole citation metadata from external sources can only be displayed in structured format. Since the reference extraction process (see Section 4.2) also yields structured metadata for both electronic and scanned print documents, the structured reference list view is also available for these cases.

4.3.3 Metadata Editing. So far, the editorial system is intended to be the only front end for the LOC-DB. Thus, it needs to cover a set of certain classical operations of a database management system [12]. These operations comprise a minimum of CRUD operations for all considered data structures. Not only the links between resources, i.e. citations need to be inserted, but also the resources' metadata occasionally needs to be updated by the librarian. To this extent, dedicated forms targeting the editing process are necessary. On top of the four basic operations which directly use the services provided by the back end (see Section 4.1.1), we can enhance the

user experience by offering interfaces for more complex operations. For instance, consider the functionality of committing a citation link between two resources. The librarian has selected an entry of the reference list in the source resource and is about to commit the link to the selected target resource. By design, the target resource might come from either an internal suggestion, an external suggestion or from the extracted metadata. When the resource was not present in the internal database before, a new resource gets automatically created. Potential changes to metadata of resources coming from external suggestions are kept locally in the editorial system until a citation link pointing to this specific resource is committed. In case the reference entry was pointing to another resource before, not only the link itself but also the higher-level `cites` property of the citing resource is updated.

On the whole, we created an editorial system comprising the functionality of data ingestion, reference linking, along with both mandatory and convenient database management operations. An interactive demo of the front end is available online[26] and the source code is available openly on GitHub[27].

5 USER STUDY AND QUANTITATIVE EVALUATION

As our final goal is to answer the question whether libraries could manage to catalog citations when provided with the right tools and processes, a constant evaluation of all software components and steps in our workflow is needed. To obtain a full picture, we use a mixed method design [10]. More specifically, we perform a qualitative user study as well as a quantitative evaluation.

5.1 User Study

Central to early-stage development of the LOC-DB platform are formative evaluations with our stakeholders [10]. Here, we applied several methods including prepared mock-ups and early user interface prototypes, and discussed them in interview sessions with domain experts. In addition, we have conducted a public workshop of LOC-DB for information professionals.[28] The domain experts acknowledged the complexity of resolving the citation link between two bibliographic resources and agreed on the abstraction of a bibliographic resource that disregards concrete embodiments for the purpose of storing citation data. They highlighted that the three tasks of scanning print documents, cataloging the document's metadata and resolving ambiguities are typically performed by different person groups. For resolving a citation link, the domain experts feel comfortable with receiving suggestions from the union catalogs and pointed out that these suggestions help to minimize the time required for each citation link, which is in turn crucial for the overall feasibility. In the podium discussion of the public workshop, there was consensus that citation data is a new, but important task for libraries. Questions about who should be responsible for storing and distributing the data were discussed as well as how to integrate the information into current systems and data formats [21].

[26] https://locdb.bib.uni-mannheim.de/demo-frontend/
[27] https://github.com/locdb/locdb-frend
[28] https://locdb.bib.uni-mannheim.de/blog/en/workshop-2/

Table 2: Comparison between the Text-driven vs. the Layout-driven Reference Extraction Method [1].

Method	Total References Extracted	Extracted %age
Text-driven	3,645	71.7%
Layout-driven	4,323	84.9%

5.2 Quantitative Evaluation

For the quantitative evaluation, we constantly monitor the whole process and collect usage data. The most important questions that can be answered in a quantitative way is how much time executing our workflow for indexing citations actually takes. In order to answer this question, we focused on finding appropriate measures for three major steps in our current process.[29]

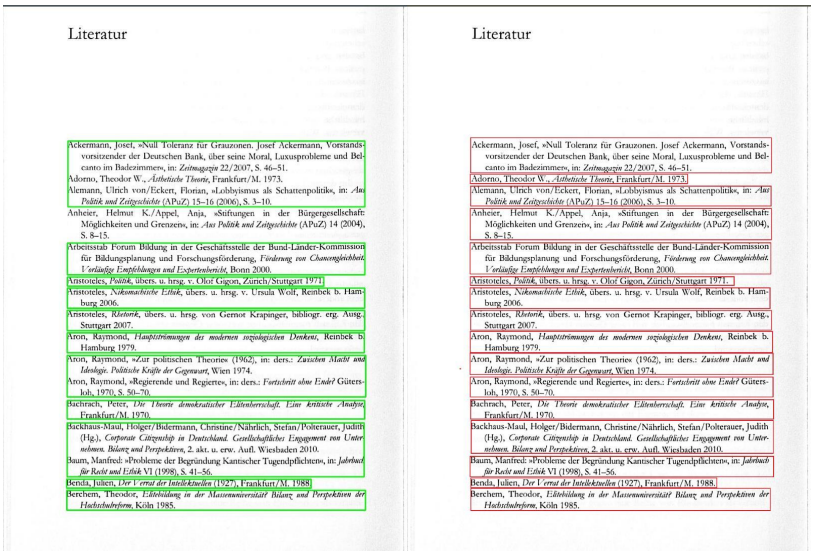
5.2.1 Scanning. In case we are dealing with print resources, the reference pages need to be scanned (see above). As part of our study, we measured the amount of pages a student assistant can scan using standard library scanners. We found that scanning can be done at a rate of at least 100 pages per hour per person, largely depending on the person. This is of course really slow, but it gives us an upper bound of around 15 minutes for scanning from an average book with 26 pages of references (Table 1). More importantly, we found that the additional scanning time, when done by student assistants whenever time permitted it, did not significantly affect other processes in the library. This is hard to measure, but we estimate that the additional scanning prolongs the processing of a book on average by only 3 minutes.

5.2.2 Automatic Reference Extraction. The quality of the automatic extraction of references is crucial for the overall performance of our system. The better the reference extraction, the easier it is for our users to find correct matches and the more efficient is the linking process in the end. Therefore, we evaluate the text-driven vs. the layout-driven reference segmentation we presented above. In order to do so, the same image was processed through both approaches. Figure 4a and Figure 4b represent the reference detection output from the text-driven and the layout-driven approach respectively.

It can be clearly observed that the text-driven approach was unable to detect some references and has also classified multiple references as a single reference. On the other hand, the layout-driven approach worked very well and, at least in this example, it detected all references clearly, despite of the fact that it did not take any textual information into account.

A comprehensive evaluation [1] of both approaches with a total of 5,090 references from 286 scanned documents (Table 2) showed that the layout-driven segmentation performed almost 13 percentage points better than the text-driven approach, i.e., ParsCit, which is used in many projects applying automatic citation extraction.

[29] We also plan to add more metrics regarding other questions, such as the completeness and quality of the generated data. For now, we assume human-level quality as the data only gets stored when the librarians checked it for correctness and completeness.



(a) Text-driven Detection Output **(b) Layout-driven Detection Output**

Figure 4: Comparison between Text-driven and Layout-driven Reference Detection using a Sample Reference Page.

5.2.3 Reference Linking. One of the most valuable insights we are trying to obtain from the project is, which features of our infrastructure must be improved to decrease the reference linking time. The goal is that in the end we can ensure optimal support of the user in the reference linking step.

To this end, we added a logging mechanism to our graphical user interface and to our back end and defined a series of events which we log in order to trace how the user is interacting with the system. These events, e.g. the user selected a resource or the suggestions from external sources arrived, allow us to calculate several measures, which we define as follows.

- Citation Linking Time: The overall time needed to link a reference of a source bibliographic resource to a target bibliographic resource.
- Suggestion Retrieval Time: The amount of time needed to retrieve suggestions and display them to the user. We distinguish here between internal and external suggestions.
- Number of Searches performed per Reference: The number of searches performed in order to retrieve a matching reference.

Figure 5 visualizes all cases we obtained for citation linking. It can be seen that from 300 seconds on the times start to increase drastically, which is due to the fact that we do not (yet) have a pause button in the system. Sometimes people get distracted or forget to close the editorial system. We therefore decided to exclude all cases with a time larger than 600 seconds, to be precise: 87 cases with times up to 6 days.

Table 3 lists minimum, maximum, and median time in seconds for the remaining 444 cases that we used for this report.

On the one hand, we can see that the median reference linking time is currently about 89 seconds. This is a good starting point for a first prototype in an early stage, in which the internal data base is not populated yet. Here is clearly potential for improvement.

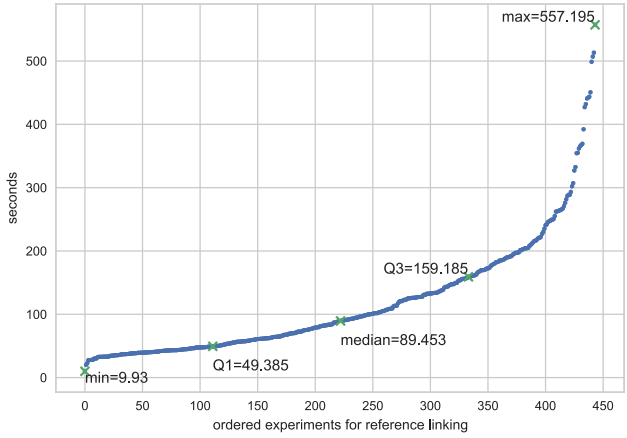


Figure 5: All 444 samples with their reference linking time ordered.

Table 3: Minimum, maximum, and median time in seconds for the reference linking step with a sample size of 444.

Criterion	Min	Max	Median
Citation Linking (s)	9.93	557.195	89.453
Internal Sug. Retrieval (s)	0.015	0.537	0.057
External Sug. Retrieval (s)	0.498	95.652	0.886
# Searches per Reference	1	36	2

On the other hand, when looking at the histogram (Figure 6), it can be seen that many links are resolved within one minute, with a minimum time of under 10 seconds. A deeper analysis showed that

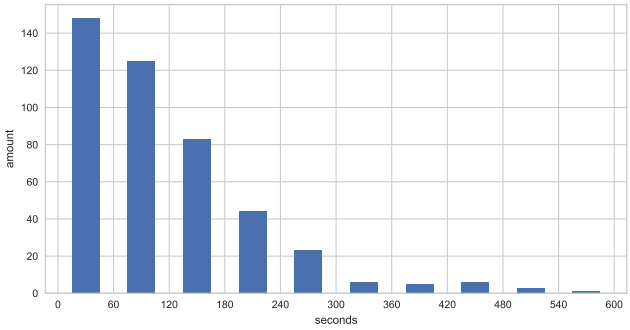


Figure 6: Histogram of reference linking times.

in those cases the metadata of the resource matching the reference was already in our internal database and that therefore the user only had to select the matching, internal suggestion. We can thus assume that the linking process will become much faster with a growing database.

Table 3 also shows the times that are needed to retrieve suggestions from internal and external sources. While the internal suggestions arrive usually within 60 ms, with a rare maximum of 10 seconds, the user has to wait significantly longer for the external ones, where the maximum time even reached 96 seconds until they arrive (average: 4 seconds).

We learn from these results that this step might slow down the user when working with our infrastructure. Therefore, we plan to pre-compute the external suggestions once we have extracted the reference strings. The user could then re-trigger the retrieval of external suggestions only if the pre-computed ones are not good enough, usually after editing the metadata. This way, the overall citation linking time can be decreased easily.

However, as the number of searches per reference indicates, currently usually the user has to search twice to get a correct suggestion. Only for 4.5% of all references, we could immediately provide the correct suggestion. This illustrates how crucial the original data extraction is for the performance. But also the matching process needs to be improved constantly to further reduce the number of searches.

6 DISCUSSION

We developed a distributed, semi-automated system to create and store citation data which is designed for usage in digital libraries. We have shown that the required additional resources for operating this semi-automated system in a distributed manner are affordable. Plus, not only digitally available resources but also scanned print documents may be processed using the LOC-DB system.

While we do not doubt that fully-automated citation extraction tools such as ParsCit [5] can be employed in many cases to get reasonable results, we believe that our semi-automated approach is doable in libraries. In contrast to the automated extraction, it guarantees a human-level quality for the curated citation data, which we deem important particularly for applications like scientometrics, where careers are at stake depending on the results.

Table 4: Estimation about the number of full-time employees needed to process all literature of social sciences bought in 2011 by Mannheim University Library, depending on the time t in seconds to resolve a reference.

t	1	5	10	20	30	60	120
employees	0.1	0.5	1	2	3	5.9	11.9

The LOC-DB system was developed in close collaboration with two libraries, namely the Mannheim University Library as well as ZBW – Leibniz Information Centre for Economics. The key requirements of the librarians could therefore be addressed early in the development process. We invited the German library community for a public workshop for evaluation of the LOC-DB system. Except for minor usability issues, the feedback considering the system itself and the general concept was thoroughly positive. We can therefore expect our findings to generalize to other digital, scientific libraries.

So, how much would it take to actually maintain LOC-DB? Based on our experience so far, we would like to ponder on some rough estimations. We deliberately chose the corpus in our project to be a complete year (2011) of incoming literature in a field where the Mannheim University Library has a strong focus, the social sciences. If LOC-DB is possible at all, it has to be possible to deal with all incoming relevant publications.

How many people would have been needed in Mannheim in 2011 to maintain LOC-DB for the social sciences? From our experience with the already scanned books we see that most books have less than 20 references on a single book page and hence we use this as the average number of references on a single book page. This leads to 271.000 references from all the books pages (cf. Table 1). Moreover, we identified 101 electronic journals in social sciences for which we want to resolve all articles published in 2011. An analysis of the subset of all journals in Crossref shows that there are 67 articles per journal on average (upper bound) and an article has 44 references on average.[30] This leads to 298, 000 references for all journal articles, so in sum we have around 570, 000 references overall which we need to resolve. Table 4 lists the number of full-time employees needed, depending on the required time to curate a reference.[31] At our current rate, Mannheim would need between 6 and 12 people just to maintain LOC-DB for social sciences. With a growing database, more openly available data and further improvements in our system, we can expect to reduce the time to at least 30 seconds. In our public workshop, we also asked the participants what they think is realistic for a library to invest in LOC-DB, with respect to people maintaining it. Answers varied, but half a full-time position was considered reasonable by many. This means: If many libraries join LOC-DB and collaborate, it should be possible to cover all the literature hold by these libraries. Some German libraries have already expressed their interest to test and possibly implement LOC-DB to improve their services.

[30] These estimations are made with the help of the Crossref API as shown in this Jupyter Notebook: https://github.com/locdb/locdb-journal-analysis/. The number of articles is based on registered DOIs per journal, which is an upper bound as sometimes DOIs are registered, but not actually used.

[31] For estimating the number of full-time employees, we assume that one employee works 200 days a year and 8 hours a day, resulting in 5, 760, 000 seconds a year.

7 CONCLUSION

Citations are a major resource in scientific discourse, information retrieval, and in bibliometrics. Recently, more and more initiatives are highlighting the importance of citations and propagating the idea of making reference data freely available.

We think that the responsibility of curating citation data and making it publicly available should be transferred to libraries in a distributed manner. The need to solely rely on citation data offered by publishers would be alleviated. This would directly affect the whole scientific community since funding and employment heavily depends on this citation data and the corresponding scientometrics.

In this paper, we presented our project Linked Open Citation Database (LOC-DB), which aims to prove that with well-designed processes and a distributed infrastructure the efficient cataloging and curation of the citation graph by librarians is possible. This way, we can overcome the current limitations in state-of-the art methods for automatic reference extraction and provide high-quality citation data integrated in the Linked Open Data cloud. In order to do so, we combine qualitative and quantitative evaluation methods. Here, we presented results from our development and public workshops as well as a first series of experiments, which focus on three important steps in our workflow: scanning reference pages of printed resources, automatically extracting references from various formats, and finally linking the references. The numbers reveal interesting insights: First of all, we have seen that scanning reference pages on standard library scanners is only a minor overhead to the daily work in the library. Secondly, we presented results of our automatic reference extraction, where we significantly improved the segmentation of references by using an image-based approach. Finally, we also analyzed the user interaction with our system in the citation linking step and showed that for many cases, the user already can efficiently find matching bibliographic resources for reference strings from internal and external suggestions. We are aware that our current evaluation is only a first step. Further studies are required and planned to obtain more detailed and reliable information about the practical applicability of our approach.

Therefore, we continuously evaluate and improve our workflow, with a focus on the reference extraction and the matching process, where we expect the highest impact on the overall performance.

While we do this, more and more citation data in the domain of social science is going to be generated by our librarians, which is going to be published as open data in the RDF-format. More data will be available thanks to all initiatives for open citations. As a final thought, we therefore would like to extend our question slightly: What if libraries cataloged everything and curated the citation graph? Let us find out.

REFERENCES

[1] Akansha Bhardwaj, Dominik Mercier, Andreas Dengel, and Sheraz Ahmed. 2017. *DeepBIBX: Deep Learning for Image Based Bibliographic Data Extraction.* Springer International Publishing, Cham, 286–293. https://doi.org/10.1007/978-3-319-70096-0_30

[2] Marshall Breeding. 2015. Future of Library Discovery Systems. *Information Standards Quarterly* 27, 1 (2015), 24. https://doi.org/10.3789/isqv27no1.2015.04

[3] Thomas M. Breuel. 2008. The OCRopus open source OCR system. In *Document Recognition and Retrieval XV, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 29-31, 2008. Proceedings (SPIE Proceedings)*, Berrin A. Yanikoglu and Kathrin Berkner (Eds.), Vol. 6815. SPIE, 68150F. https://doi.org/10.1117/12.783598

[4] Thomas M. Breuel, Adnan Ul-Hasan, Mayce Ibrahim Ali Al Azawi, and Faisal Shafait. 2013. High-Performance OCR for Printed English and Fraktur Using LSTM Networks. In *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013.* IEEE Computer Society, 683–687. https://doi.org/10.1109/ICDAR.2013.140

[5] Isaac Councill, C. Lee Giles, and Min-Yen Kan. 2008. ParsCit: an Open-source CRF Reference String Parsing Package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08).* European Language Resources Association (ELRA), Marrakech, Morocco. http://www.lrec-conf.org/proceedings/lrec2008/summaries/166.html

[6] Lee R. Dice. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology* 26, 3 (1945), 297–302. https://doi.org/10.2307/1932409

[7] Eugene Garfield. 1955. Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas. *Science* 122, 3159 (July 1955), 108–111. https://doi.org/10.1126/science.122.3159.108

[8] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. 1998. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the Third ACM Conference on Digital Libraries (DL '98).* ACM, New York, NY, USA, 89–98. https://doi.org/10.1145/276675.276685

[9] Annette Klein. 2017. Von der Schneeflocke zur Lawine: Möglichkeiten der Nutzung freier Zitationsdaten in Bibliotheken. *o-bib. Das offene Bibliotheksjournal* 4, 4 (Dec. 2017), 127–136. https://doi.org/10.5282/o-bib/2017H4S127-136

[10] Jonathan Lazar, Jinjuan Feng, and Harry Hochheiser. 2017. *Research Methods in Human-Computer Interaction.* Morgan Kaufmann.

[11] Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10, 8 (February 1966), 707–710.

[12] James Martin. 1981. Managing the data base environment. (1981).

[13] John Mingers and Loet Leydesdorff. 2015. A review of theory and practice in scientometrics. *European Journal of Operational Research* 246, 1 (2015), 1–19. https://doi.org/10.1016/j.ejor.2015.04.002

[14] Philippe Mongeon and Adèle Paul-Hus. 2016. The journal coverage of Web of Science and Scopus: a comparative analysis. *Scientometrics* 106, 1 (Jan. 2016), 213–228. https://doi.org/10.1007/s11192-015-1765-5

[15] Silvio Peroni. 2014. The Semantic Publishing and Referencing Ontologies. In *Semantic Web Technologies and Legal Scholarly Publishing.* Springer, Cham, 121–193. https://doi.org/10.1007/978-3-319-04777-5_5

[16] Silvio Peroni, Alexander Dutton, Tanya Gray, and David Shotton. 2015. Setting our bibliographic references free: towards open citation data. *Journal of Documentation* 71, 2 (2015), 253–277. https://doi.org/10.1108/JD-12-2013-0166 arXiv:https://doi.org/10.1108/JD-12-2013-0166

[17] Silvio Peroni and David Shotton. 2016. *Metadata for the OpenCitations Corpus.* Technical Report. https://dx.doi.org/10.6084/m9.figshare.3443876

[18] Silvio Peroni, David M. Shotton, and Fabio Vitali. 2017. One Year of the OpenCitations Corpus - Releasing RDF-Based Scholarly Citation Data into the Public Domain. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II (Lecture Notes in Computer Science)*, Vol. 10588. Springer, 184–192. https://doi.org/10.1007/978-3-319-68204-4_19

[19] Yvonne Rogers, Helen Sharp, and Jenny Preece. 2012. *Interaction Design - Beyond Human-Computer Interaction, 3rd Edition.* Wiley.

[20] Dario Taraborelli, Lydia Pintscher, Daniel Mietchen, and Sarah Rodlund. 2017. WikiCite 2017 report. (Dec. 2017). https://figshare.com/articles/WikiCite_2017_report/5648233 DOI: 10.6084/m9.figshare.5648233.v3.

[21] Christian Wilke and Regina Retter. 2017. Zitationsdaten extrahieren: halbautomatisch, offen, vernetzt. Ein Workshopbericht. *Informationspraxis* 3, 2 (Dec. 2017). https://doi.org/10.11588/ip.2017.2.43235

[22] Dietmar Wolfram. 2015. The symbiotic relationship between information retrieval and informetrics. *Scientometrics* 102, 3 (2015), 2201–2214. https://doi.org/10.1007/s11192-014-1479-0

ACKNOWLEDGMENTS

This work is funded by the German Research Foundation (DFG) under project number 311018540 (Linked Open Citation Database).

We thank Kirsten Jeude and Sarah Gatz from ZBW – Leibniz Information Centre for Economics for their participation and insightful comments during the development workshops, Fabian Steputath for helping with the front end development, and Sylvia Zander and Laura Erhard for extensive testing, identifying problems and helping to improve the overall system.