(Automated) Literature Analysis - Threats and Experiences

Yusra Shakeel
University of Magdeburg &
METOP GmbH, Germany
shakeel@ovgu.de

Jacob Krüger
University of Magdeburg &
Harz University, Germany
jkrueger@ovgu.de

Ivonne von Nostitz-Wallwitz[*]
University of Magdeburg &
METOP GmbH, Germany
ischroet@ovgu.de

Christian Lausberger
METOP GmbH, Germany
Christian.Lausberger@metop.de

Gabriel Campero Durand
University of Magdeburg, Germany
campero@ovgu.de

Gunter Saake
University of Magdeburg, Germany
saake@ovgu.de

Thomas Leich
Harz University &
METOP GmbH, Germany
tleich@hs-harz.de

ABSTRACT

The number of scientific publications is increasing each year, specifically in the field of computer science. In order to condense existing knowledge, evidence-based software engineering is concerned with systematic literature reviews, surveys, and other kinds of literature analysis. These methods are used to summarize the evidence on empirical studies – or approaches in general – and to identify gaps for new research opportunities. However, executing systematic review processes requires a considerable amount of time and effort. Consequently, researchers have proposed several semi-automated approaches to support and facilitate different steps of such methods. With our current research, we aim to assist researchers to efficiently and effectively execute different steps, namely the search for and selection of primary studies. In this paper, we report several issues we identified during our research that threaten any kind of literature analysis and hamper suitable tool support. We further recommend solutions to mitigate these threats. Overall, our goal is to raise researchers' and publishers' awareness regarding several potential threats on literature analysis, to support software engineers in designing suitable tools for research, and to encourage the research community to solve these threats.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**;

KEYWORDS

lessons learned, systematic literature review, threats to validity, software engineering, literature analysis

ACM Reference Format:
Yusra Shakeel, Jacob Krüger, Ivonne von Nostitz-Wallwitz, Christian Lausberger, Gabriel Campero Durand, Gunter Saake, and Thomas Leich. 2018. (Automated) Literature Analysis - Threats and Experiences. In *SE4Science'18: SE4Science'18:IEEE/ACM International Workshop on Software Engineering for Science , June 2, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3194747.3194748

1 INTRODUCTION

The number of scientific research articles published in computer science, particularly empirical studies in software engineering, is increasing every year [34, 38]. Consequently, a challenge for the research community is to efficiently identify and synthesize existing knowledge regarding a specific research area. For this purpose, especially systematic literature reviews have emerged as a useful method for literature analysis ever since first guidelines have been adopted for software engineering. The most prominent example may be the guidelines by Kitchenham and Charters [26]. Although systematic literature reviews have gained popularity among evidence-based researchers, the main challenges of this methodology – namely, the associated time and effort – still exist [17, 20].

To overcome these challenges, researchers have proposed strategies to semi-automate different phases of a systematic literature review [30, 31, 36, 39]. Any systematic literature review usually involves a large number of potentially relevant articles to answer the defined research questions [16]. Thus, researchers in different areas, such as, medicine, social sciences, and software engineering, focus on developing tools to facilitate the execution phase of systematic literature reviews [20, 35]. This phase comprises the most crucial steps of identifying, selecting, and assessing primary studies. Any semi-automatic approach to assist reviewers in efficiently and effectively performing these steps would be useful to reduce time and efforts, to facilitate replications, and to avoid flaws. Consequently, implementing such approaches is considered to be crucial by the software engineering community [20]. However, corresponding tools are still in development and during such projects additional threats and flaws in existing systems are revealed.

During our ongoing projects in this regard [14, 30, 38, 39], we identified issues that hamper especially the *automation* of literature

[*] This author previously published as Ivonne Schröter.

SE4Science'18, June 2, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5748-7/18/06…$15.00
https://doi.org/10.1145/3194747.3194748

analysis. In this paper, we discuss the following issues (numbers refer to the corresponding section):

4.1 Inconsistent search models and query options of digital libraries;
4.2 Limited possibility to crawl libraries to retrieve results;
4.3 Limited access to bibliographic information of articles;
4.4 Conditional access to full texts of research articles;
4.5 Inconsistencies when exporting search results from digital libraries; and
4.6 Differences in the formatting (i.e., author names).

For each of these issues, we (1) *describe the problem statement*, (2) *discuss potential threats*, and (3) *propose solutions*. Additionally, we explain for each threat how we identified it and aim to provide an example. We remark that not all threats can be addressed by researchers while performing a systematic literature review or implementing a corresponding tool. Some are connected to publishers and digital libraries, who could ideally aim to resolve them, too.

Nonetheless, researchers have to be aware of these threats for multiple reasons, the most prominent being: Firstly, they should aim to avoid or resolve the threats when performing a literature analysis or developing a corresponding tool. Secondly, if they cannot resolve the threats, researchers should at least be aware that these exist and discuss them. Finally, any existing and future systematic literature review or other literature analysis faces these threats and may be biased, due to the applied processes and tools. We hope that our work helps to raise the awareness of such threats and initiates further tool development to support the research community.

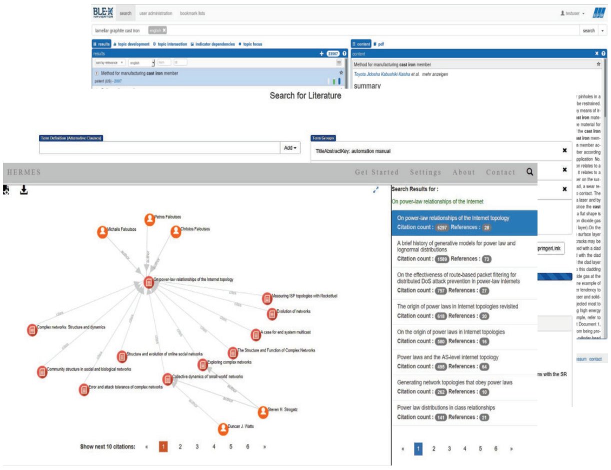
2 LITERATURE ANALYSIS

A literature analysis is a secondary study that enables researchers to obtain insights into prior work in a research topic and to establish foundations for future works [11, 42]. Despite its importance, performing an efficient and effective literature analysis is usually challenging, as it comprises costly tasks and several forms have established: systematic literature reviews have emerged as a distinct, well-defined method to analyze literature and to answer research questions in a systematic manner [5]. Similarly, mapping studies outline existing literature regarding a specific concept to identify gaps in the current research [9]. A literature survey summarizes and presents conclusions by technically reviewing large amounts of recently published scholarly articles. In the context of this paper, we refer to each of these types as literature analysis and rely on systematic literature reviews as specific example.

To identify relevant studies, reviewers select appropriate digital libraries, specific journals, or conferences. Mostly, automated keyword searches in digital libraries provide an initial set of results. However, digital libraries vary in size, scope, and supported features, as they are usually maintained by different organizations or academic institutions. Evidence-based researchers commonly rely on established databases, such as, the ACM Digital Library, Scopus, Google Scholar, IEEE Xplore, SpringerLink, and ScienceDirect.

3 TOWARDS AUTOMATED ANALYSIS

To reduce the time and effort required for literature analysis and systematic literature reviews in particular, researchers ask for supporting tools. Fortunately, the evidence-based research community is actively working on this area and proposes different semi-automatic approaches. However, Hassler et al. [20] determine that there is still a lack of tool support for systematic literature reviews – with facilitating the selection process of studies being one of the most desired features by the community.


BLE-X NAVIGATOR
search
user administration
bookmark lists
testuser
lamellar graphite cast iron
english
search
results
topic development
topic intersection
indicator dependencies
topic focus
content
pdf
23567
sort by relevance
Search for Literature
Term Definition (Alternative Clauses)
Add
Term Groups
HERMES
Get Started
Settings
About
Contact
Petros Faloutsos
Michalis Faloutsos
Christos Faloutsos
On power-law relationships of the Internet topology
Measuring ISP topologies with Rocketfuel
Evolution of networks
Complex networks: Structure and dynamics
A case for end system multicast
The Structure and Function of Complex Networks
Structure and evolution of online social networks
Exploring complex networks
Community structure in social and biological networks
Collective dynamics of 'small-world' networks
Error and attack tolerance of complex networks
Steven H. Strogatz
Duncan J. Watts
author
cites
Show next 10 citations: « 1 2 3 4 5 6 »
« 1 2 3 4 5 6 »

Method for manufacturing **cast iron** member
patent (US) - 2007

Method for manufacturing **cast iron** member
Toyota Jidosha Kabushiki Kaisha et al. mehr anzeigen
summary

TitleAbstractKey: automation manual

Search Results for :
On power-law relationships of the Internet

On power-law relationships of the Internet topology
Citation count : 6297 References : 28

A brief history of generative models for power law and lognormal distributions
Citation count : 1589 References : 73

On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets
Citation count : 797 References : 27

The origin of power laws in Internet topologies revisited
Citation count : 618 References : 20

On the origin of power laws in Internet topologies
Citation count : 580 References : 16

Power laws and the AS-level internet topology
Citation count : 495 References : 64

Generating network topologies that obey power laws
Citation count : 262 References : 10

Power law distributions in class relationships
Citation count : 141 References : 21

Figure 1: User interfaces of tools developed during our previous and ongoing research [14, 30, 38].

Any literature analysis involves vital tasks, such as, identifying and selecting relevant studies [17, 38]. To enable users to perform these steps in a more feasible manner, different tools can be used – but a single tool to support the entire process is still missing. With our ongoing research [14, 30, 38, 39], we aim to contribute towards automating literature analysis by developing approaches that assist reviewers during various phases. Currently, we develop individual components for these phases. We show current user interfaces of some tools we developed during our ongoing research in Figure 1. For example, the displayed screens list results of search queries, compute several metrics, and perform graph analysis to show literature networks. These components will later be combined into a single tool – or can be reused by others – for performing literature analysis with the least possible manual effort. Our goal is not only to support researchers and practitioners to conveniently analyze literature in the context of systematic literature reviews, but in any type of literature analysis.

Our current research includes empirical studies on systematic literature reviews and the needs of software engineers. Consequently, some initial results have emerged: Based on the response of industrial partners [38], we developed user interfaces and metrics that consider context information to rank literature. We are implementing a semi-automatic approach to search articles in different digital libraries, paying attention to their limitations and adapting the search processes accordingly [30]. Moreover, we are building a graph database application to support scholarly network analysis [14]. Finally, we are working on metrics to select primary studies and assess their quality [39]. These approaches are not limited to systematic literature reviews, but are useful for any type of literature analysis. In the future, we will refine, extend, and integrate these components into one tool, also supporting customizations for different analysis processes.

During the aforementioned works, we encountered a number of issues. Some of these must completely or partly be resolved by the publishers and administrators of digital libraries, for instance, the consistency of search models (Section 4.1) and providing the possibility to crawl bibliographic information with application programming interfaces (Section 4.2). Additional issues occur when implementing metrics for articles, as exported citations from different libraries are not standardized, which can result in mismatches and faulty computations (Section 4.6). With this paper, we aim to report to the software engineering community that such problems threaten the validity of a literature analysis and hamper tool development that is crucial for researchers.

4 THREATS

In this section, we describe the issues of automating literature analysis we faced while developing our tools. We provide a detailed explanation of each issue, exemplifying when it arises, the problem statement, and resulting threats to validity. Furthermore, we discuss possible solutions that can be applied by publishers and researchers to overcome these threats. At this point in time, we are aware of the following six issues. Each corresponds to one or more threats and they are closely related or even overlapping, making a clear separation challenging.

4.1 Search Models

Any literature analysis – and systematic literature reviews in particular – should rely on a structured, unbiased search strategy to retrieve all relevant articles for answering the defined research questions [27]. A systematic search strategy specifies the search string, data resources, search method, and parts of an article to search in, such as, title, abstract, and keywords [15]. Evidence-based researchers usually select various data resources to identify important literature that may be relevant to their topic of interest. We investigate the search requirements of four digital libraries that are frequently used by software engineering researchers [30], namely: IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink. The former two libraries contain the most important journals and conferences in computer science [25, 45]. However, due to their high relevance and frequent usage, we also test the requirements of ScienceDirect and SpringerLink.

We analyze the search models of the selected libraries in order to compare their functionalities. In Table 1, we provide an overview of the supported search fields of these libraries. The most frequently supported search fields, in addition to the full text, are the title and keywords. Such refinements allow for searches of important terms to be conducted only within the specified part of an article. Additionally, we inspect the search model specified for each library, as this determines how relevant articles are found.

Problem Statement – An obstacle during the implementation of an efficient search strategy, is the use of different search models and query operators to identify relevant articles from various digital libraries. Within the scope of this work, we highlight two problems concerning the search fields and term requirements specified by libraries. We observe a lack of uniformity in the features supported by libraries, which can negatively impact the consistency, quality, and completeness of search results.

Table 1: Overview of the search fields provided in regularly used digital libraries.

Search Field	IEEE Xplore	ACM DL	ScienceDirect	SpringerLink
Full text	✓	✓	✓	✓
Title	✓	✓	✓	✓
Abstract	✓	✓	✓	✗
Keywords	✓	✓	✓	✓
Title-Abstract-Keyword	✗	✗	✓	✗

Threat 1: Inconsistent Search Fields. We find that there are variations in the search field options provided by different digital libraries. Without specifying such search fields, almost all libraries that we examined apply a full text search. As we see in Table 1, most libraries also support separate searches in title, abstract, and keywords, although SpringerLink being an exception, as it does not offer to search in abstracts. Consequently, using a search query on a combination of digital libraries that support different search fields may not be consistent. For example, if for an analysis reviewers aim to focus on abstracts for their search, this will cause problems when using SpringerLink.

The combination of the aforementioned search fields is sometimes provided as a single function, which we refer to as title-abstract-keyword search. This function searches for occurrences of the search string in all three decisive areas of an article. We can see in Table 1 that, apart from ScienceDirect, none of the examined digital libraries provides a separate title-abstract-keyword search feature. This combination is useful to accurately find the most potentially relevant articles, as it examines the most important parts of an article that include a summary and indexing information. Despite its importance and regular usage, a single function for this combination is still not applicable for a literature analysis investigating libraries other than ScienceDirect.

Solution – To overcome this threat of inconsistent search fields, search strings must be adapted to each library while considering the described constraints. Although an initial examination of each digital library is challenging, it seems necessary to ensure the uniformity of searches among them. A more efficient solution would be beneficial, as the necessary time increases and the liability for errors rises. Here, it would be optimal if publishers synchronized their search models in this regard. Still, researchers can carefully plan their searches in advance to prevent unnecessary adaptations during the search. Considering the title-abstract-keyword search, we can partly overcome this problem by combining the single fields in one query. However, the resulting length of the search string can cause additional problems, as several libraries are limited in the query length they allow (cf. following threat).

Threat 2: Inconsistent Syntax and Filters. We also observe a lack of consistent search term requirements for different digital libraries. To ensure accuracy of results across different libraries, researchers need to alter the search syntax and keywords based on the varying requirements of each library. For example, Google Scholar only supports terms up to 256 characters [22] and provides limited filtering capabilities to their users, such as, refinements

regarding the scientific area or year [33]. Furthermore, Soni and Kodali [40] as well as Abdelmaboud et al. [1] detect problems using long search terms in the ACM Digital Library, IEEE Xplore, Emerald, and SpringerLink – a problem we also experienced regularly. Such limitations in using different libraries greatly affect the search strategy, as an appropriate combination of search terms must be defined for each library. This requires a detailed examination of individual libraries, which is also challenging on its own, resulting in an even more difficult and time consuming search process.
Solution – To meet this problem, we started to develop a concept that automatically adapts search terms to different libraries and, thus, facilitates the search [30]. Still, there is need for more effective automated approaches to address this threat and make unified searches possible for reviewers who want to select different digital libraries for their literature analysis. An alternative solution is to use digital libraries that index multiple publishers, such as, Google Scholar or Scopus, and apply snowballing [43] on the results. However, this is also connected to several other threats we explain in this paper. Overall, the most desired but arguably least likely approach would be a unification process between publishers.

4.2 Crawling Libraries

Crawling is necessary to enable tools to automatically search and analyze digital libraries. It is enabled through a web-bot – a simple program to analyze web pages – that searches for hyperlinks and organizes a series of page requests on these links. For security reasons and to guarantee fair resource sharing, crawling is not encouraged by digital libraries, such as, Microsoft Academic and ACM Digital Library, resulting in blocked servers or financial costs. To still provide access for tooling and allow partially automated searches, some digital libraries offer application programming interface functionality to their users.

An application programming interface allows sharing of content between software applications and is applicable in a variety of contexts [12]. For example, such interfaces are used for embedding content from one website to another or interacting with data in a programmatic manner. There are few documentations of application programming interfaces that are provided by digital libraries, allowing users to determine the range of functions and syntax for formulating search terms. The response to a request is in raw format and no further conversion of data is required [30]. Using application programming interfaces for searches broadens the opportunity for users to retrieve a large number of relevant results that are available in digital libraries.

Problem Statement – During our works, we experienced some limitations in the provided application programming interfaces. To determine their capabilities for advanced search functionality, we investigate the digital libraries we selected earlier. Some of them allow searches of their freely available content through an application programming interface for noncommercial use. These are, along with the corresponding URLs:

- IEEE Xplore (http://ieeexplore.ieee.org/gateway/)
- ScienceDirect (http://api.elsevier.com/content/search/scidir)
- SpringerLink (http://api.springer.com/meta/v1/json)

However, it must be noticed that application programming interfaces are limited and not provided by all digital libraries. This is a serious limitation in the context of literature analysis. In addition, as the same search models are applied and results provided in the application programming interfaces, the threats from Section 4.1 and Section 4.6 can also apply to them.

Threat: Missing support of application programming interfaces. The concept of using application programming interfaces to search a library is still missing in some commonly used digital databases – for example, the ACM Digital Library that hosts one of the largest bibliographic collections with a focus on the field of computer science. Due to their limitations of application programming interface search facility, reviewers performing literature studies are unable to investigate such libraries as a search venue. Especially, as crawling the websites themselves results in temporal blocking, this issue significantly hampers tool development.
Solution – To address this threat, it is preferable that administrators of digital libraries provide well-documented application programming interfaces to their users. This could encourage the research community to improve the available tools in this regard and analyze the existing literature in more detail. As we mentioned earlier, consistency in the search functionalities offered by various digital libraries is beneficial to achieve an unbiased search procedure and facilitate tool support.

4.3 Snowballing

An initial set of relevant articles is identified by applying a search query on selected search venues. The selected primary articles form the basis to identify further relevant ones using their meta-data and citation relationships, such as, referenced and citing articles, which we sketch in Figure 1 and Figure 2. This snowballing [43] step is important to retrieve relevant articles that may have been overlooked (e.g., from other venues, due to synonyms) in the initial search phase. Especially, if the concept is hard to define or the existing literature is quite large, further steps should be taken to ensure recognition of all relevant articles [7]. For this reason, forward and backward snowballing is usually performed, which are time-consuming tasks if done manually.

It is also possible to consider related articles by searching through relationships other than citations [14]. For example, considering articles by researchers that collaborate frequently with the author of an important primary one. This way, snowballing can be framed as a specific case of a more generic search strategy that relies on scholarly network analysis. Such strategies consist of traversing through the networks in which articles are embedded, such as, collaboration networks, to find related ones.

Problem Statement – An initial search usually provides an incomplete set of the relevant literature. Thus, identifying related articles is recommended. For an automated approach to identify related articles, it is essential to retrieve all their meta-data. However, this is only partially possible, due to restrictions imposed in digital libraries. Additionally, digital libraries often provide insufficient support for users to explore relationships between articles that go beyond a limited number of citations. Most of them only present a certain number of articles (cf. Table 2) for a query, wherefore it is even technically impossible to fully apply snowballing. As a consequence, manual and automatic approaches to identify related articles require lots of effort.

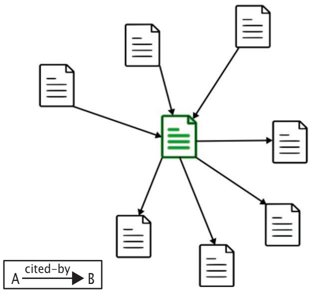


Figure 2: Concept of snowballing to identify related articles using citations and references, based on Lausberger [30].

Threat 1: Missing Support for Snowballing. Due to missing automation for forward snowballing, most researchers tend to exclude it from their search process [30]. For instance, Gami et al. [19] is cited by 1,754 articles that are practically impossible to assess without any tool support. As a consequence, literature analyses are often performed without considering citing articles. For example, the search process followed by Kitchenham et al. [24] involves a manual search of specific search venues followed by using experts' opinions to select articles. Similarly, studies conducted by Kitchenham et al. [28] and Zahedi et al. [44] follow a process that includes automated keyword search but exclude forward snowballing. If reviewers choose to eliminate snowballing from their selection process, they are probably neglecting some relevant articles. Thus, there is a potential threat to the validity and accuracy of the literature analysis. We believe that for a conclusive study, it is essential to identify all available literature – for which an initial search is insufficient.

Solution – As we mentioned earlier, manually determining literature based on citation relationships requires a significant amount of time and effort. Despite its importance for a complete literature analysis, an automatic approach to support reviewers is still missing. For backward snowballing, some solutions exist, for example, to parse the articles [29] or rely on citation graphs of publishers [14]. Still, while these approaches can help to some extent, we see the need to further improve such approaches, either by corresponding tools or by publishers providing the necessary meta-data. In a limited scope, screening all citing and referenced articles is possible and should be applied.

For large collections, we require a database structure that manages meta-data and citation links to enable tools to automatically crawl these. Here, articles can be recognized by using digital object identifiers. Still, to this end, access to the meta-data of related articles must be provided by the administrators of digital libraries. To access citing articles, application programming interfaces can be used for some digital libraries, for instance, ScienceDirect. However, currently it is only possible to obtain the number of citing articles, which is not sufficient. It would be helpful if digital libraries would enable access to bibliographic information to allow tools to crawl these consistently. Otherwise, it may be necessary to create an additional network that stores these information but is separated from the publishers' libraries. For example, Microsoft provides suitable snapshots of the indexed articles in their Academic Graph.

Threat 2: Missing Network Information. Scientific publications are embedded in complex networks of collaboration, citations, co-citations, bibliographical coupling, institution/publication venue dependencies, and topics [14, 38]. Exploring these networks in a systematic way can be crucial, providing researchers with an assurance of recall. While digital libraries support the exploration of citation networks to some extent, there is almost no support for considering co-citations or other networks. As a result, valuable data for relevance analysis is not available.

Solution – Digital libraries should provide standardized support and access methods to networks of indexed articles, similar to the Microsoft Academic Graph. This would support reviewers in considering better heterogeneous networks related to articles. This may also require significant efforts in developing standards along with efficient techniques for indexing and searching through these networks. Again, a database structure that separately maintains such networks may be a way to go from a researcher's perspective.

4.4 Full Text Analysis

For an accurate literature analysis, critically reviewing the full text of articles is essential. Such an assessment enables reviewers to precisely determine articles' relevancy for the concerned research area. Consequently, this aspect is significant for the final selection of articles, as well as for extracting direct evidence to answer the defined research questions. One of our studies indicates an inadequacy of tools acknowledging full text analysis to select and assess articles [39]. Most of the proposed automated approaches consider only specific parts of the article, such as, title, abstract, and keywords [2, 41]. Interpreting the full text of articles is a requirement for precisely evaluating the available information, either manually or by employing an automated procedure.

Problem Statement – During our research, we observe some limitations concerning the access to existing scientific research. These include the unavailability of complete articles, restricted access imposed by publishers, and missing tools to support assessments of full texts. Such problems must be addressed to ensure completeness, validity, and repeatability.

Threat: Availability of Articles. The problem of articles being unavailable to some researchers, is one of the well-known obstacles for performing a literature analysis and research in general. This problem occurs if only an abstract of the article has been published, due to language barriers, or simply because publishers prohibit access for some researchers. Consequently, some research contributions remain unavailable for the scientific community, which is a problem for the authors and reviewers alike. For example, IEEE Xplore and the ACM Digital Library permit complete access to their publications only to paying users [13]. These usually include researchers affiliated with certain institutions or members willing to pay for their subscription. This means that scientific researchers are not provided the same level of access to the available literature [10], which threatens especially the demanded repeatability of systematic literature reviews [6].

Solution – It is important to resolve this issue whether the literature analysis is performed automatically or manually. Generally, for research articles that are not available, reviewers may contact the authors to ask them for an authors' version. However, this procedure can be time consuming or even infeasible for situations where the authors are no longer actively involved in research. Furthermore, automation cannot be achieved this way. Due to the conflict of interest between researchers, publishers, and funding agencies for providing access to research articles, we can hardly propose a suitable solution ourselves. Instead, we recommend to carefully document which articles may be relevant but could not be included because of unavailability.

4.5 Exporting Citations

A structured search query provides reviewers with a set of results. To perform a further analysis, the articles' citations are exported from the search venues into an appropriate format or reference tool. Some of the most common bibliographic data formats are *BibTex* – based on the TeX typesetting program and LaTeX macros – and *CSV* – a tabular format accepted by most databases and spreadsheets. For our research, we examine the digital libraries listed in Table 2, to determine their facilities for retrieving bibliographic data.

Besides the data format, we also inspect the maximum number of results and the possibility to download content, such as, abstract and full text. Through our analysis, we observe inconsistencies in the information made available by digital libraries. This poses a barrier to efficiently export citations from different libraries and automate literature analysis.

Problem Statement – A problem with retrieving bibliographic data is the variation of features that digital libraries offer. There are different constraints, making it difficult for reviewers to perform comparable searches in multiple libraries [3]. For example, the maximum number of citations that can be exported from digital libraries varies, as we show in Table 2. Furthermore, there is a lack of standardization in the bibliographic data format. Advanced exporting functionalities, such as direct export to reference management programs (e.g., Mendeley or RefWorks), are supported by few digital libraries. Moreover, some also allow to download information other than the publication details, such as, keywords and abstracts, to perform an analysis. However, not all the digital libraries support the same features, for instance, SpringerLink only allows CSV exports of citations.

Threat 1: Limited Number of Exportable Citations. The results obtained through a search query are sorted by the digital libraries according to their relevancy. We can observe in Table 2 that all considered libraries impose a limit to the number of exportable – partly even to the number of viewable – articles. Thus, the bibliographic data of articles, beyond a certain limit, is unachievable even if the user manually selects them. Among the digital libraries we selected, SpringerLink and ScienceDirect enable the users to retrieve bibliographic information of up to 1,000 citations. IEEE Xplore and the ACM Digital Library allow users to save the meta-data for a larger range of citations. Still, without any specific selection made by the user, the libraries rate the relevancy by predefined measures and, thus, bias which articles appear first – and can be seen. This limitation could be relatively significant, if a search query results in a larger number of articles than the defined limit. In such circumstances, there may be the possibility that a proportion of relevant studies is excluded during this stage of the process.

Solution – To accurately select relevant articles from the search results, users analyze their bibliographic data. A preferable solution, provided by the administration of digital libraries, is enabling users to retrieve all citations obtained from a search query. This can help to ensure that all relevant articles are considered in a literature analysis. However, if a limit must be set, then digital libraries may standardize it. A larger number of articles being shown and exportable would be preferable to include all relevant works and information. Considering researchers, we recommend to use snowballing to overcome the limits of articles that can be exported. Additionally, we recommend to potentially redefine the search query, as it may be too broad.

Threat 2: Inconsistencies in Exportable Formats. We also notice a variation in the bibliographic data format offered and tools supported by digital libraries for exporting citations. IEEE Xplore, ACM Digital Library, and ScienceDirect allow users to export citations in two commonly used file format: CSV and BibTex. However, ScienceDirect uses a different BibTex format and SpringerLink provides only CSV exports. Thus, for an automated approach to analyze search results from different libraries, it is required to incorporate various formats for performing further analysis based on the bibliographic data. Unfortunately, there is currently also no reference management tool that is supported by all digital libraries.
Solution – A uniform format to export bibliographic information is important for efficient literature analysis. The ability to directly export citations into reference management tools, such as, RefWorks or Mendeley, allows users to manage a large collection of studies more efficiently. As there is no other common format than CSV and no tool directly supported by each library, managing the articles of a literature analysis is challenging. To overcome such limitation, the ideal solutions would be that the administrators of digital libraries provide consistent options for exporting citations. Again, this may be unlikely and we are currently aiming to develop parsers that can import different formats to unify them for a set of reference managers.

Threat 3: Inconsistencies in Exportable Data. We noticed a similar problem considering the differences in bibliographic data that can be exported from digital libraries. For example, IEEE Xplore and ScienceDirect enable users to export abstracts along with meta-data of articles. In contrast, the ACM Digital Library and SpringerLink lack this feature, making it impossible to assess the abstracts of articles using an automated approach. Some digital libraries also enable users to download the full text of articles to which they have access, but limit the number of articles that can be downloaded at once. For example, IEEE Xplore and ScienceDirect allow full text downloads of 10 and 25 articles at a time, respectively.
Solution – To improve automated literature analysis, it is beneficial to obtain information other than citations. Consequently, it would be helpful if digital libraries provide consistent and extended functionalities to access different information that are available to a researcher. At the moment, we are aiming to at least automate the extraction of all publicly accessible data provided by the available application programming interfaces. Such information are often posted on the article's website and usually include citations, abstracts, keywords, and references.

Table 2: Properties of selected libraries for exporting search results.

Digital Library	Limit	Formats								Contents		
		CSV	Plain Text	BibTex	RIS	RefWorks	Mendeley	ACM Ref	EndNote	Citation	Abstract	Full Text
IEEE Xplore	2000	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
ACM DL	2000-2300	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗
ScienceDirect	1000	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓
SpringerLink	1000	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

Table 3: Author-name formats for selected libraries.

Digital Library	Author-name format
IEEE Xplore	B. A. Kitchenham
ACM DL	Kitchenham, Barbara
ScienceDirect	Barbara Kitchenham
SpringerLink	Barbara Kitchenham

4.6 Different Formatting

The meta-data of a research article comprises important publication details, such as, document title, author, year, publisher, venue, and keywords. Comprehensive and accurate meta-data – especially author names – is essential to identify and assess a research article [18]. Previous research has shown that abbreviations, contractions, omissions, typos and alternations occur in published datasets of digital libraries, such as, the ACM Digital Library, DBLP, and Citeseer, with abbreviations and omissions accounting for more than half of all variations in names [37]. To automatically assess and relate articles, consistency in the such formats (especially author names) is an important factor.

Problem Statement – For our analysis, we examine citations exported from the aforementioned libraries. We inspect the author-name format, focusing on the use of initials and order of full names, as represented by each of them. Any difference in the format makes it more challenging to perform a literature analysis – especially when using an automated methodology. Same authors may not be recognized, due to difference in their name-format provided by the digital libraries. We present our observations in Table 3, where we use the author name: *Barbara Ann Kitchenham*, as an example to illustrate our analysis.

Threat: Inconsistent Formatting. For an automated methodology to assess articles based on the authors, inconsistent author names within different digital libraries is a problem. Depending on the metrics that are used, such inconsistencies can result in considerable different outcomes. Additionally, it becomes difficult to directly account articles to authors to evaluate their work as a whole. The example we show in Table 3 illustrates such different formats. ScienceDirect and SpringerLink support similar author-name formats, while IEEE Xplore and the ACM Digital Library offer different ones. During our investigation, we also notice that such inconsistent formats even exist within single libraries.

Solution – To handle this problem, publishers need to standardize the formats used in their articles. While this is already done for some aspects, for example, the ACM Computing Classification System as a well-defined ontology, the format of other parts is completely inconsistent. We exemplified this for the author names but it also applies to keywords or titles, especially if the data is exported from the library into a reference manager. As a complete consistency check by the publishers seems unlikely and some faults are also caused by researchers, we are aiming to develop an approach to compare terms (e.g., with similarity metrics) and classify them to provide a unified view on the included articles.

5 RELATED WORK

Especially evidence-based software-engineering researchers are investigating the efforts and challenges of performing literature analysis. Still, Marshall and Brereton [32] determine that most proposed tools lack effectiveness, are in their early stages, or face significant limitations. Their findings reflect the immaturity of this research area and laid foundations for future work, encouraging researchers to address this deficit.

Carver et al. [11] identify that *searching digital libraries*, *selecting papers*, and *extracting data* are the aspects that reviewers find most difficult, time consuming, and ask to support with tools. Another study conducted by Imtiaz et al. [23] also implies that defining a search strategy, using various digital libraries, as well as planning and extracting data are the most challenging tasks. Furthermore, the obstacles associated with tooling are highlighted by Hassler et al. [21] and are similar to those we experience and explain in Section 4.1 and Section 4.5. Other researchers also highlight limitations of automating literature analysis [3, 4, 8, 13].

All these works emphasize the need for research tools that support literature analysis. While there are some minor overlaps in the threats we report and such works, we provide more detailed insights, a consolidated overview, and real-world examples. More importantly, we propose solutions to address the identified threats and are developing corresponding tools. Thus, our contributions in this paper are complementary to such works.

6 CONCLUSION

In this paper, we elaborate a number of threats that hamper automated literature analysis and recommend possible solutions to address them. We identified these threats during our ongoing research to implement an efficient and effective tool for any literature analysis and systematic literature reviews in particular. Our goal is to improve the understanding and awareness for these threats, which are mainly concerned with:

- Inconsistencies of digital libraries, in terms of search fields, syntax requirements, exportable citations, and their lack of support for advanced search facilities;
- Insufficient support of snowballing;

- Unavailability of scientific articles; and
- Variations in bibliographic data formatting in digital libraries.

We believe that the issues highlighted in this work can be resolved by publishers and researchers. This way, the research community can be encouraged to become involved in proposing further tools to automate literature analysis. Thus, along with reporting threats, we also discuss some possible solutions on which we are partly working. These serve as basis for further research and motivate our current engineering activities. Additionally, we hope that the findings presented in this paper help researchers conducting, reviewing, and reading literature analysis – providing them with helpful insights to see threats they may face.

Acknowledgments Supported by DFG grants LE 3382/2-1 and SA 465/50-1, and the DAAD STIBET Matching Funds grant.

REFERENCES

[1] Abdelzahir Abdelmaboud, Dayang N. A. Jawawi, Imran Ghani, and Abubakar Elsafi. 2015. A Comparative Evaluation of Cloud Migration Optimization Approaches: A Systematic Literature Review. *Journal of Theoretical & Applied Information Technology* 79, 3 (2015), 395–414.

[2] Ramon Abilio, Flávio Morais, Gustavo Vale, Claudiane Oliveira, Denilson Pereira, and Heitor Costa. 2015. Applying Information Retrieval Techniques to Detect Duplicates and to Rank References in the Preliminary Phases of Systematic Literature Reviews. *CLEI Electronic Journal* 18, 2 (2015), 1–24.

[3] Muhammad A. Babar and He Zhang. 2009. Systematic Literature Reviews in Software Engineering: Preliminary Results from Interviews with Researchers. In *ESEM*. IEEE.

[4] John Bailey, Cheng Zhang, David Budgen, Mark Turner, and Stuart Charters. 2007. Search Engine Overlaps: Do They Agree or Disagree?. In *REBSE*. IEEE.

[5] Sebastian K. Boel and Dubravka Cecez-Kecmanovic. 2015. Debating Systematic Literature Reviews (SLR) and their Ramifications for IS: A Rejoinder to Mike Chiasson, Briony Oates, Ulrike Schultze, and Richard Watson. *Journal of Information Technology* 30, 2 (2015), 188–193.

[6] Sebastian K. Boell and Dubravka Cecez-Kecmanovic. 2015. On Being 'Systematic' in Literature Reviews in IS. *Journal of Information Technology* 30, 2 (2015), 161–173.

[7] Andrew Booth, Anthea Sutton, and Diana Papaioannou. 2016. *Systematic Approaches to a Successful Literature Review*. SAGE.

[8] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. Lessons From Applying the Systematic Literature Review Process Within the Software Engineering Domain. *Journal of Systems and Software* 80, 4 (2007), 571–583.

[9] David Budgen, Mark Turner, Pearl Brereton, and Barbara A. Kitchenham. 2008. Using Mapping Studies in Software Engineering. In *PPIG*. PPIG.

[10] Alex Byrne. 2003. Digital Libraries: Barriers or Gateways to Scholarly Information? *The Electronic Library* 21, 5 (2003), 414–421.

[11] Jeffrey C. Carver, Edgar Hassler, Elis Hernandes, and Nicholas A. Kraft. 2013. Identifying Barriers to the Systematic Literature Review Process. In *ESEM*. IEEE.

[12] Cesare Concordia, Stefan Gradmann, and Sjoerd Siebinga. 2010. Not Just Another Portal, Not Just Another Digital Library: A Portrait of Europeana as an Application Program Interface. *International Federation of Library Associations and Institutions Journal* 36, 1 (2010), 61–69.

[13] Oscar Dieste, Anna Grimán, and Natalia Juristo. 2009. Developing Search Strategies for Detecting Relevant Experiments. *Empirical Software Engineering* 14, 5 (2009), 513–539.

[14] Gabriel Campero Durand, Anusha Janardhana, Marcus Pinnecke, Yusra Shakeel, Jacob Krüger, Thomas Leich, and Gunter Saake. 2018. Exploring Large Scholarly Networks with Hermes. In *EDBT*. ACM.

[15] Tore Dybå, Torgeir Dingsøyr, and Geir K. Hanssen. 2007. Applying Systematic Reviews to Diverse Study Types: An Experience Report. In *ESEM*. IEEE.

[16] Katia R. Felizardo, Stephen G. MacDonell, Emilia Mendes, and José C. Maldonado. 2012. A Systematic Mapping on the Use of Visual Data Mining to Support the Conduct of Systematic Literature Reviews. *Journal of Software* 7, 2 (2012), 450–461.

[17] Katia R. Felizardo, Norsaremah Salleh, Rafael M. Martins, Emília Mendes, Stephen G. MacDonell, and José C. Maldonado. 2011. Using Visual Text Mining to Support the Study Selection Activity in Systematic Literature Reviews. In *ESEM*. IEEE.

[18] Anderson A. Ferreira, Adriano Veloso, Marcos A. Gonçalves, and Alberto H.F. Laender. 2010. Effective Self-training Author Name Disambiguation in Scholarly Digital Libraries. In *JCDL*. ACM.

[19] Apoor S. Gami, Brandi J. Witt, Daniel E. Howard, Patricia J. Erwin, Lisa A. Gami, Virend K. Somers, and Victor M. Montori. 2007. Metabolic Syndrome and Risk of Incident Cardiovascular Events and Death: A Systematic Review and Meta-Analysis of Longitudinal Studies. *Journal of the American College of Cardiology* 49, 4 (2007), 403–414.

[20] Edgar Hassler, Jeffrey C. Carver, David Hale, and Ahmed Al-Zubidy. 2016. Identification of SLR Tool Needs - Results of a Community Workshop. *Information and Software Technology* 70 (2016), 122–129.

[21] Edgar Hassler, Jeffrey C. Carver, Nicholas A. Kraft, and David Hale. 2014. Outcomes of a Community Workshop to Identify and Rank Barriers to the Systematic Literature Review Process. In *EASE*. ACM.

[22] Muhammad Ilyas and Siffat U. Khan. 2015. Software Integration in Global Software Development: Success Factors for GSD Vendors. In *SNPD*. IEEE.

[23] Salma Imtiaz, Muneera Bano, Naveed Ikram, and Mahmood Niaz. 2013. A Tertiary Study: Experiences of Conducting Systematic Literature Reviews in Software Engineering. In *EASE*. ACM.

[24] Barbara A. Kitchenham, Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic Literature Reviews in Software Engineering - A systematic literature review. *Information and Software Technology* 51, 1 (2009), 7–15.

[25] Barbara A. Kitchenham, David Budgen, and Pearl Brereton. 2015. *Evidence-based Software Engineering and Systematic Reviews*. CRC Press.

[26] Barbara A. Kitchenham and Stuart Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE-2007-01. Keele University.

[27] Barbara A. Kitchenham, Tore Dybå, and Magne Jørgensen. 2004. Evidence-Based Software Engineering. In *ICSE*. IEEE.

[28] Barbara A. Kitchenham, Rialette Pretorius, David Budgen, Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. 2010. Systematic Literature Reviews in Software Engineering – A Tertiary Study. *Information and Software Technology* 52, 8 (2010), 792–805.

[29] Pavel Laskov and Nedim Šrndić. 2011. Static Detection of Malicious JavaScript-Bearing PDF Documents. In *ACSAC*. ACM.

[30] Christian Lausberger. 2017. *Konzeption von Suchprozessen und Suchstrategien für Systematische Literatur Reviews*. Master's thesis. Otto-von-Guericke-University Magdeburg. In German.

[31] Viviane Malheiros, Erika Höhnr, Roberto Pinho, Manoel Mendonca, and José C. Maldonado. 2007. A Visual Text Mining Approach for Systematic Reviews. In *ESEM*. IEEE.

[32] Christopher Marshall and Pearl Brereton. 2009. Tools to Support Systematic Literature Reviews in Software Engineering: A Mapping Study. In *ESEM*. IEEE.

[33] Gideon Mbiydzenyuy. 2015. Arrival Times with Hours of Service Regulations for Truck Drivers-Tracks and Gaps from Current Research. In *ITSC*. IEEE.

[34] Matthew R. McGrail, Claire M. Rickard, and Rebecca Jones. 2006. Publish or Perish: A Systematic Review of Interventions to Increase Academic Publication Rates. *Higher Education Research & Development* 25, 1 (2006), 19–35.

[35] Jefferson S. Molleri and Fabiane B. V. Benitti. 2012. Automated Approaches to Support Secondary Study Processes: A Systematic Review. In *SEKE*.

[36] Fábio R. Octaviano, Katia R. Felizardo, José C. Maldonado, and Sandra C. P. F. Fabbri. 2015. Semi-automatic Selection of Primary Studies in Systematic Literature Reviews: Is it Reasonable? *Empirical Software Engineering* 20, 6 (2015), 1898–1917.

[37] Byung-Won On, Gyu S. Choi, and Soo-Mok Jung. 2014. A Case Study for Understanding the Nature of Redundant Entities in Bibliographic Digital Libraries. *Program* 48, 3 (2014), 246–271.

[38] Ivonne Schröter, Jacob Krüger, Philipp Ludwig, Marcus Thiel, Andreas Nürnberger, and Thomas Leich. 2017. Identifying Innovative Documents: Quo Vadis?. In *ICEIS*. ScitePress.

[39] Yusra Shakeel. 2017. *Supporting Quality Assessment in Systematic Literature Reviews*. Master's thesis. Otto-von-Guericke-University Magdeburg.

[40] Gunjan Soni and Rambabu Kodali. 2011. A Critical Analysis of Supply Chain Management Content in Empirical Research. *Business Process Management Journal* 17, 2 (2011), 238–266.

[41] Federico Tomassetti, Giuseppe Rizzo, Antonio Vetro, Luca Ardito, Marco Torchiano, and Maurizio Morisio. 2011. Linked Data Approach for Selection Process Automation in Systematic Reviews. In *EASE*. IET.

[42] Jane Webster and Richard T. Watson. 2002. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *Management Information System Quarterly* 26, 2 (2002), xiii–xxiii.

[43] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *EASE*. ACM.

[44] Mansooreh Zahedi, Mojtaba Shahin, and Muhammad A. Babar. 2016. A Systematic Review of Knowledge Sharing Challenges and Practices in Global Software Development. *International Journal of Information Management* 36, 6 (2016), 995–1019.

[45] He Zhang and Muhammad A. Babar. 2010. On Searching Relevant Studies in Software Engineering. In *EASE*. BCS.