Exploring Automatic Search in Digital Libraries – A Caution Guide for Systematic Reviewers

Paramvir Singh
National Institute of Technology
Jalandhar, India
singhpv@nitj.ac.in

Karanpreet Singh
National Institute of Technology
Jalandhar, India
karanpreet.13.cse@nitj.ac.in

ABSTRACT

Search phase is considered as one of the most important steps in conducting secondary studies such as systematic literature reviews and mapping studies. In recent times, automatic search in digital libraries and academic search engines has been the preferred method of search phase execution for most software engineering related secondary studies. However, there are no previous studies that report or evaluate the secondary study relevant search features of these electronic data sources. We perform a feature analysis (screening mode) based evaluation of five widely used digital libraries (IEEE Xplore, ACM DL, SpringerLink, ScienceDirect and Wiley) in terms of their respective features required to support the search phase of secondary studies. We identify a total of 68 search related features and conduct a comprehensive exploration into their execution behaviors. The overall work presents a useful caution guide for systematic reviewers who plan to use the identified features for executing various search phase steps of their secondary studies.

KEYWORDS

Systematic review, mapping study, automatic search, digital libraries, evidence based software engineering, search strategy

1 INTRODUCTION

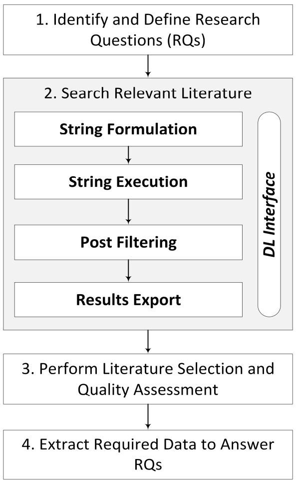
Secondary studies such as Systematic Literature Reviews (SLRs) and Systematic Mapping Studies (SMSs) are well-known methodologies supporting Evidence Based Software Engineering (EBSE). Both these dominant EBSE methodologies are strongly dependent on a systematic review protocol and the underlying search process. Where the manual search is considered useful for searching through a restricted set of sources, automatic search is suitable when the research is aiming at completeness (and scale) for specialized topics (e.g. in SLRs) and broad topics (e.g. in SMSs). Hence, most of the SLRs and SMSs published in the last decade have relied on the automatic search method.

Although quite reliable, the results of the automatic search could lead to false positives/negatives because of either a wrong selection (or concatenation) of search keywords, or bugs in the search related features of the digital libraries employed. The first problem is generally handled by conducting a *pilot study* to validate the search keywords and strings [1]. The second problem is however much more difficult to counter. Some of the well-known critical works [2, 3] on application of SE SLRs and SMSs also warn researchers against relying too much on digital libraries for identifying relevant literature. Although researchers have been employing manual methods (tough and error-prone, especially when conducting large scale studies) for validating the automatic search results, we believe a comprehensive understanding about the actual search behavior of digital libraries beforehand could help their cause to a great extent.

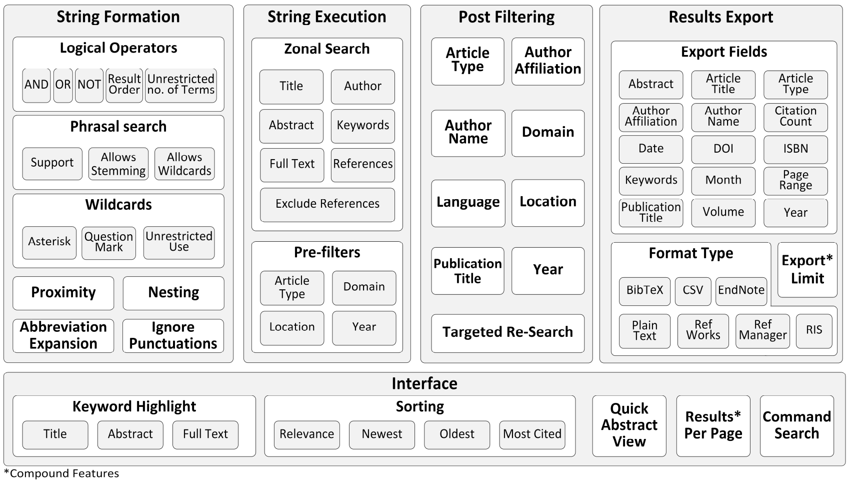
In this work, we conduct a qualitative feature analysis based evaluation of five Digital Libraries (DLs), namely IEEE Xplore (IEX), ACM DL (ACM), SpringerLink (SPL), ScienceDirect (SCD) and Wiley, which are well-known to support SE related SLRs and SMSs. The rationale behind selecting these five candidate DLs is derived from the electronic sources of relevance for SE researchers as suggested in [4] and [5], and the widespread use of these DLs in the studies analyzed to draw the motivations for this work (Section 2). The work was conducted between Dec. 1, 2016 and Jan. 31, 2017. Throughout the rest of this text, we use the term Systematic Review (SR) to refer both SLR and SMS.

The major contributions of this work are: (i) extraction and classification of a set of digital library search features relevant to SRs in general and SE SRs in particular (Section 3); (ii) qualitative analysis of the support provided by all candidate DLs for the identified feature sets (Section 4); and (iii) a set of high-level recommendations (for SR researchers) derived from the qualitative analysis results (Section 5). Overall, this work presents a useful caution guide for SR researchers planning to use the candidate DLs for implementing their search processes. At the same time, this work provides an opportunity for the developers of these DLs to improve various features which are found lacking in dealing with search string execution. To the best of our knowledge, there are no previous studies that explore and analyze the SR related search features of the selected DLs.

Paper Preprint 2017



(a)



(b)

Figure 1: (a) Subtasks of an SR automatic search phase, (b) Feature set classification.

2 MOTIVATIONS

This work has been motivated by the need to understand thoroughly the DL peculiarities that are vital for carrying out various subtasks of the automatic search step of a typical SR review process (see Fig. 1a). Another motivation is the fact that most of the SRs report incomplete and misleading information relevant to the formation and execution of search strings with respect to the DLs used. To support our motivation, we conducted a broad study on a set of 26 recently published SRs (found through Google Scholar) in reputed SE research journals. We targeted only the recent studies (2016-2017) to insulate this analysis from the effects of any recently introduced changes in DL search features. For each of these SRs, we manually extracted the relevant information [6] to draw a number of observations.

About 46% studies do not use dedicated strings for individual DLs used in the search process. Only 27% studies actually enlist customized search strings used for each DL. For 42.4% studies, at least one of the reported search strings fails when executed in respective DLs. Out of 93 string-DL combinations across all 26 studies, 11.3% show errors and 10.3% show doubtful number of results. Only approx. 8% studies mention search execution inconsistencies across DLs as a possible threat in their respective ‘Threats to Validity’ sections (also evident from [7]). Only 12% studies directly or indirectly mention/discuss about the characteristics of any string formation features such as stemming, proximity, wildcards, concatenation, word limit, etc.

There are many little feature specific behavioral characteristics attached with these DLs that have the potential to atleast impact the construct and conclusion/replication validity of the SRs. Also, many of these characteristics are not mentioned in the ‘help/documentation’ sections of these DLs, and are completely subject to the researchers’ experience with the respective features.

3 EVALUATION METHOD

We used a qualitative evaluation method called feature analysis (in screening mode) defined under DESMET methodology [8] to evaluate various candidate DLs. Feature analysis aims at qualitatively assessing the level of support that is essentially required for a particular task/activity [8]. A typical screening mode evaluates a method or a tool based on the literature instead of actually experimenting its use [8]. However, as this work involves evaluations targeting the inconsistencies linked to the actual execution behaviour of SR related features of selected DLs, we extend the screening mode toward an initial *experimentation* by adding practical search executions to it.

3.1 Feature Classification and Scoring Criteria

The finalized feature set hierarchy containing five different classes and 68 individual features is illustrated in Fig. 1b. The identified features are divided into two broad categories, namely simple and compound, on the basis of respective feature values. The judgment score to a simple feature is assigned as 5 or 0 based on whether a candidate DL supports a specific simple feature or not. Whereas, the actual feature values are normalized between the range 1 and 5 to compute the judgment score of a compound feature. A total of 66 simple and 2 compound (shown by '*' in Fig. 1b) lowest-level features were identified. Every feature is then assigned an importance level using an ordinal scale comprising four distinct levels namely mandatory (M), highly desirable (HD), desirable (D), and nice (N) [9]. These levels are quantified by their respective weight values, which are decided as 1 for nice, 3 for desirable, 6 for highly desirable, and 10 for mandatory [10]. Weight value of a feature set (parent feature) is computed by summing the weight values of each of its sub-features. Weighted Score (*WS*) value is calculated by multiplying judgment score and weight value of a feature. For instance, if a highly desirable (weight value of 6) feature is supported by a DL, then its *WS* is 6×5, i.e. 30. Finally, we compare DLs based on the Overall Scores (*OS*) of five top level feature sets computed using (1) [11].

$$OS = \frac{\sum_{i=1}^{n}(W_i PS_i)}{\sum_{i=1}^{n}(W_i)} \quad (1)$$

where, W_i is the weight value and PS_i is the relative percentage score of the i^{th} sub-feature in a feature set.

4 FEATURE SETS AND ANALYSIS RESULTS

We found it appropriate to describe various features in tight integration to their analysis information with regards to the candidate DLs. In order to study the execution behavior of selected features, we shortlisted a set of common keywords and search terms [6] extracted from search strings used in previous SRs related to the field of SE. These keywords were used to form the desired search strings required to assess the performance of various features. Further, except for the *zonal search*, all other experimental (test) searches were restricted to the article titles in order to manage the study scope. The search strings that were actually fed into the DLs for experimentation are represented as "*SS: [Search String]*" [6]. Following sub sections briefly discuss feature support analysis results for the identified feature sets with respect to the candidate DLs. Due to space limitation, Table 1 only presents the *WS* values of *String Formation* feature set and its underlying features. The detailed feature support information along with the corresponding *WS* scores generated for all five feature sets could be found at [6].

Table 1: Support (SP) and Weighted Scores (WS) of String Formation feature set

Feature	Importance	IEX		ACM		SPL		SCD		WLY	
		SP	WS	SP	WS	SP	WS	SP	WS	SP	WS
Logical Operators											
AND	M	√	50	√	50	√	50	√	50	√	50
OR	M	√	50	√	50	√	50	√	50	√	50
NOT	HD	√	30	√	30	√	30	√	30	√	30
Result Order	D	√	15					√	15	√	15
Unrestricted No. of Terms	M		0	√	50	√	50	√	50	√	50
Total Weighted Score			145		180		180		195		195
Wildcard											
Asterisk	M	√	50	√	50	√	50	√	50	√	50
Question Mark	D		0		0	√	15	√	15	√	15
Unrestricted Use	D		0	√	15	√	15	√	15	√	15
Total Weighted Score			50		65		80		80		80
Phrasal Search											
Support	M	√	50	√	50	√	50	√	50	√	50
Allows Wildcard	HD		0		0	√	30	√	30	√	30
Allows Stemming	HD		0	√	30		0	√	30	√	30
Total Weighted Score			50		80		80		110		110
Proximity	HD	√	30		0	√	30	√	30		0
Nesting	M	√	50	√	50	√	50	√	50	√	50
Abbreviation Expansion	N		0		0		0		0		0
Ignore Punctuations	HD	√	30	√	30	√	30	√	30	√	30

4.1 String Formation

A single search string may comprise several simple and complex search terms. Syntactical components supported by individual DLs help determine the structure of a search string. Various features belonging to *String Formation* set are discussed below.

4.1.1 Logical Operators. According to our analysis, all three logical operators (AND, OR, and NOT) are implemented by the five candidate DLs. ACM also supports making use of AND, OR, and NOT operators by overriding the default query, and applying the required modifications to the syntax. In SPL, the logical operators do not work in zonal (title) search from the advanced search page in SPL. Interestingly, AND operator behaves as a normal search keyword in this case.

The search strings used in some of the SRs failed to execute in IEX during our motivational analysis [6] due to the restrictions on the number of search terms allowed (max 15) per search string. Apparently, a typical query length based on our motivational analysis of 26 SRs is 14.84. This calls for an increase in the allowed search string length limit in IEX. Also, we found that the 15-term string limit in IEX is not free of inconsistencies. For more than one instance during our experimentation, search strings exceeding 15 keywords connected using logical operators (AND/OR) worked without any problems.

Brereton et al. [2] observed that the listing of articles in the search output was dependent on the order of search terms. To verify this, we conducted an experiment on the candidate DLs using two of the search strings used in [10], and found that the article order in the search result was identical for both the search strings in IEX, SCD, and WLY but different for ACM and SPL.

4.1.2 Wildcards. Wildcard characters can significantly reduce the number of keywords required to gather the relevant literature. Where asterisk ('*' → zero or more characters) is supported by all the DLs, question mark ('?' → exactly one character) is supported by SPL, SCD, and WLY. IEX forbids the use of more than five wildcard characters in a given search string. A number of previous SRs (e.g. [12]) reported using more than 5 wildcards in their IEX search strings, which clearly presents a conflicting scenario.

4.1.3 Phrasal Search. DLs let a user search for word phrases by enclosing them in double quotes. We found that all DLs fully support the phrasal search. This was validated after successfully querying phrases of word lengths greater than one. Nonetheless, phrasal search restricted to *Article Title* in WLY (advanced search) yielded erroneous results. For instance, *SS: "software lines"* and *SS: software lines* delivered the same number of results. Such unexpected glitches might lead to superfluous result output. It is therefore recommended to validate the compatibility between phrasal search and zonal search (*Article Title* in this case) by analyzing the respective search results on different DLs, before actually finalizing the string.

When searching for a specific phrase, it sometimes becomes essential to incorporate its inflected variations in the search string. For instance, a user would prefer obtaining results for both "software measurement" and "software measures", when querying "software measure". The compatibility of phrasal search with stemming and wildcards is highly desirable as it significantly reduces the search string length and complexity. ACM, SPL, and WLY hold full compatibility of phrasal search with stemming, i.e. stemming of all the keywords comprising a phrase is automatically supported. However, the inclusion of stem variants is purely based on a DL's stemming efficiency (Section 4.6). To cope up with such an issue, a user can utilize wildcards to portray any number of characters. Then again, wildcards in a phrase (enclosed in quotes) are only supported in SPL, SCD, and WLY.

4.1.4 Proximity. When simple concatenation of two keywords, for example *SS: dynamic AND metrics*, yields a large number of unwanted results, proximity operators can be highly productive by enabling a localized search that targets two keywords within a specified distance. This feature is not often utilized but can be very effective if one or more keywords are likely to occur between two search keywords. For instance, it is better to concatenate keywords "dynamic" and "metrics" using ONEAR/2 to locate phrases like "dynamic coupling metrics" and "dynamic object oriented metrics", and eliminate instances where "dynamic" and "metrics" occur at a distance greater than two words. Proximity searches are executed using NEAR/n and ONEAR/n operators in IEX and SPL; and w/n and pre/n in SCD.

4.1.5 Nesting. A search string can comprise a number of sub strings connected using different logical operators. Every such sub string is usually enclosed within a pair of parentheses that not only improves the readability but also defines its execution precedence among other parts of the string. We verified successful nesting support up to deep levels in all candidate DLs by executing experimental search strings [6].

4.1.6 Abbreviation Expansion. The lack of automatic abbreviation expansion in all DLs forces a user to include the corresponding full forms in addition to the abbreviations. During our motivational analysis, we noticed that the authors have used phrases like *"Open Source Software"* and *"Capability Maturity Model"* along with their abbreviations, OSS and CMM respectively [13], [14]. The presence of such a feature would highly benefit a user to formulate shortened search strings. In cases where the same abbreviation expands to multiple phrases belonging to different domains, concatenating an appropriate keyword can control the scope of search results.

4.1.7 Ignore Punctuations. DLs usually ignore punctuations like '-' and '_' in search strings. For instance, *SS: run-time* is treated as *SS: run AND time* during the search process. Notably, this will return articles containing "run-time" and "run time", but not "runtime", which is an appropriate word in this context. DLs also ignore punctuations from phrases enclosed in quotes e.g. *SS: "run-time"* works exactly as *SS: "run time"*. SCD offers a strict form of phrasal search using curly braces wherein no punctuations are ignored, and hence exact phrases are searched.

4.2 String Execution

For controlling the result output, DLs provide several constraint parameters that can be applied to a search before actually executing a query.

4.2.1 Zonal Search. Zonal search facilitates probing the selected parts of an article such as title, abstract, keywords, etc. We explored only the most used zonal search features in previous SE SRs. Every DL permits restricting the search to either article title, author name, or full text. SPL allows zonal search based on only title or full text. SCD and WLY provide additional ability of restricting the search to article's reference text (bibliography). Such a feature might aid in accelerating backward snowballing. All DLs except SCD incorporate reference text in their full text search. This is important because a search execution may result in a large number of irrelevant results if the reference text is included in the search.

4.2.2 Pre-Filters. Pre-filters are another way of controlling the search output by limiting the search scope prior to executing the search strings. Article pre-filtering based on their respective years of publication has been implemented by all candidate DLs. Another pre-filter option that displays results for a particular article type (conference article, journal article, magazine, etc.) is implemented by IEX and SCD. When performing a manual search, pre-filtering based on the conference location becomes desirable. Such an option is only available in ACM. Often the search output is clogged with articles belonging to domains other than the ones anticipated. SCD brings forth a pre-filter option for restricting the article lookup within a chosen domain.

4.3 Post Filtering

Where the pre-filters are applied before string execution, post-filters primarily screen the result output on the basis of specified criteria supported by the DLs. Sometimes, while formulating and validating a search string, it is desirable to find a set of articles within the search results. Out of the five candidate DLs, such a provision is only offered by IEX. The option to filter results based on the publication year is supported by all but WLY. SPL and SCD also allow screening based on the subject domain to effectively get rid of articles belonging to unwanted subjects. SPL provides a filtering option to display articles written in selected languages. IEX and ACM provide filtering based on author names or affiliations, which can significantly ease manual search. Various other filtering options can be found at [6].

4.4 Results Export

After executing the search string and performing requisite filtering, the results need to be exported from DLs for subsequent SR operations. Each of ACM, SPL, and SCD support exporting first 1000 search results, whereas IEX exports first 2000 results. In order to extract results beyond these limits, a filtering based strategy targeting one of the possible options such as year, article type, etc. could be employed to export the results in phases. The support for exporting search results in WLY is limited to a single page. As a result, the maximum number of results that can be exported at once relies on its page length (20 for WLY).

4.4.1 Format Type. A variety of file formats are used to export the search results generated by individual DLs. CSV and BibTeX are two of the most preferred export formats. IEX, SPL, and ACM support CSV export format, whereas all candidate DLs except SPL support BibTeX export format. Notably, CSV export format option is available in IEX only when exporting first 2000 results. It is noteworthy that in order to export the search results in a format other than CSV, a user has to make manual selection of individual articles from the search output (which is again limited to a page) before exporting. Information about various other export formats is available at [6].

4.4.2 Export Fields. Several types of information like publication year, title, volume number, DOI, etc. are associated with each article found in the search results. The information pertaining to each article that is necessary for an SR must be contained in the file exported by DLs. Each candidate DL supports exporting article information containing fields such as year, article title, publication title, authors, volume number, etc. Information on other export fields is available at [6].

4.5 Interface

Apart from all the features discussed above, various other DL features that can affect an SR have been collated under *Interface* feature hierarchy.

4.5.1 Sorting. After the execution of a search string, the order in which the search results are listed can be controlled by choosing from the sorting options provided by each DL. By default, the search results are ordered by their relevance to the search string. Sorting based on the citation count is only supported by IEX and ACM. This is especially helpful in SMSs where the importance of the studies is decided by the related number of citations [15]. Additional sorting option can be found at [6].

4.5.2 Keyword Highlight. It is always helpful for a user if the searched keywords are distinctively highlighted in each result on the search output page. IEX highlights keywords in both title and abstract of search results. SCD highlights keywords only in result titles. Unexpectedly, SPL highlights keywords only in abstracts but does not extend this feature to the article titles. WLY does not support any type of keyword highlighting in search results. ACM provides a comprehensive result view by displaying selected portions from the full text (including reference text) of the article containing the searched keywords.

4.5.3 Quick Abstract View. While conducting a manual search, it can be highly convenient for a user to have abstract text presented along with each of the articles listed in the result output. IEX and SCD provide the complete abstract text within the search results. SPL contains a link corresponding to each article in the result output that displays a popup window showing first two pages (containing abstract) of the article. Each search result in ACM shows only first 2-3 lines of the complete abstract. Therefore, it does not qualify for this feature.

4.5.4 Results Per Page. This feature refers to the maximum number of search results accommodated on a single page. Few results per page would force a user to frequently navigate to successive pages for retrieving subsequent results. Hence, higher the number of results per page, easier the manual search iterations. SCD and IEX permit a maximum of 200 and 100 results respectively. All other DLs limit the same to 20.

4.5.6 Command Search. DLs also provide a command line feature to execute long and complex search strings. IEX, ACM, and SCD have a command line textbox usually accessible from the advanced search or expert search options. SPL and WLY do not explicitly provide such dedicated command line textboxes but nonetheless users can execute their complex queries through primary search textboxes. Consequently, every DL does fulfill this feature directly or indirectly.

4.6 Stemming

In the context of DLs, stemming refers to the process of reducing a particular keyword of a search string to its root form and then comparing it with the keyword's inflected variants. For instance, if the searched keyword is "validates" then the words like "validating", "validated", etc. should also return a positive match to the search, as all of them share the common root form "validate". DLs usually perform automatic keyword stemming, thus cutting down the efforts required in manual accommodation of keyword variations during the design of a search string. The search algorithms in DLs commonly look for plural and basic verb forms of a given keyword. We selected a few root words and their inflected forms commonly used in SE domain [6]. A search restricted to the title was then conducted using the chosen keywords to assess the performance of automatic stemming in the candidate DLs.

IEX, SCD, and WLY possess almost identical stemming capabilities, where only the plural form, and the verb forms ending with 'ing', 'ed' were recognized. There was a minor difference in the result counts of different word forms in WLY (e.g. Programs - 37549, Programming- 37576), which can be ignored. ACM outperforms every other DL in terms of finding all the variants of selected root words. Although SPL search documentation does indicate a support for automatic stemming, in practice, stemming does not seem to be supported in SPL as indicated by our experimental results [6]. We recommend SR authors to utilize wildcards instead of relying on automatic stemming to avoid error prone results.

5 SUMMARY AND RECOMMENDATIONS

A comparison of all the top level DL feature sets in terms of their overall score *(OS)* values is shown in Fig. 2. Overall, considering all five feature sets, IEX and SCD provide the most comprehensive support to the conduct of search process during SRs. Whereas, SPL and WLY need to improve significantly in enhancing their support for the selected feature sets. Based on the qualitative feature analysis of DLs, we make a number of high level recommendations to the researchers targeting to use these DLs for their prospective SRs as follows. Firstly, it is important to *define customized strings* for each DL, as the implementations and hence the behaviors of SR relevant features vary across different DLs. Secondly, we recommend a *thorough learning of feature-level expected/observed behavioral aspects* of each DL (as presented in Section 4) before forming the dedicated initial SR search strings. Thirdly, a *feature specific validation study* for each search string defined for each DL during the SR planning phase should be performed before committing to finalized search strings. Understanding various DL features precisely would help avoid missing any relevant study during the search process, and hence improve the reliability of automatic search. Various DL inconsistencies (snapshots can be found at [6]) related to the

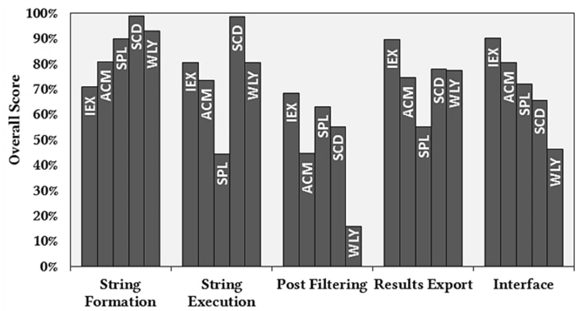


Figure 2: Overall Score (OS) of feature sets.

search process of SRs must be discussed in the 'Threats to Validity' section of these studies under 'Construct and Conclusion Validity'.

6 STUDY LIMITATIONS

We might have missed some of the SR relevant features/feature sets in defining our feature classification. However we made sure that we closely follow the SR review protocols as suggested in [1, 4], and those followed by various SE SRs in order to learn required search related DL features before coming up with the final feature set. As far as the feature completeness is concerned, we did not cover all word stemming possibilities due to the sheer scale of such a task. A validity threat is also posed due to the positioning of each feature under various top level feature sets. The feature level evaluation outcomes for each DL are subject to change because the developers might be working on a specific DL search inconsistency issue as we finalize this paper. We hence maintain the state of DL features that are relevant to SRs, in the form of a GitHub repository [6]. It is possible that a wrong impression about a specific feature is registered. This threat is mitigated by repeating each experimental search 10 times. We did not use the complete search strings as required for a qualitative experiment. We mitigate this threat by exercising selected words from general SE research domain to maximize the coverage of most stemming options.

7 RELATED WORK

The previous works related to this study include a number of tertiary studies, and studies targeting the evaluation of DLs for various purposes, led by SRs especially pertaining to the field of SE. Chen et al. [16] and Bailey et al. [17] suggested an evidence based methodology to select the DLs for conducting SRs. These works analyzed DL contributions and result overlaps to compare the performance of a number of DLs from the perspective of SE secondary studies. They however do not perform any feature analysis based evaluation of the feature sets required for the execution of the SR search process. A number of studies comparing the search strategies required for SE SRs have been previously published. Zhang and Babar [5] conducted a survey on state-of-the-art SR search strategies and proposed an approach toward designing optimal search strategies for SE SRs. Ghafari et al. [18] developed a unified search tool to enhance the search process of SE SRs. Dieste et al. [19] conducted a study toward devising optimal search strategies using recall and precision parameters. In the fields other than SE, Sampson et al. [20] provided evidence based practical guidelines for assisting the review process of search strategies used in SRs.

8 CONCLUSIONS AND FUTURE WORK

This paper reports a feature analysis (screening mode) based qualitative evaluation of five DLs with an aim to support the SE researchers engaged in conducting SRs. The support for respective features as provided by the candidate DLs was iteratively validated through a set of search experiments. A set of high-level recommendations for an effective use of various DL features were drawn from the feature specific behavior and respective inconsistencies (if any) observed during the search executions of this validation process.

The five candidate DLs considered in this work only comprise an initial list of electronic databases, which we plan to extend in future to include more comprehensive academic search engines (Scopus, Web of Science, etc.) and other DLs as well. We intend to maintain the feature updates relevant to SRs in the form of a GitHub project. We will also be contacting the SR researchers to record and understand their experiences related to DL inconsistencies. Conducting a similar feature analysis in formal experiment mode is another future work.

REFERENCES

[1] B. Kitchenham and S. Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. School of Computer Science and Mathematics, Keele University, EBSE-2007-01.

[2] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. 2007. Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Sys. and Soft.*, 80 (4), 571–583.

[3] S. MacDonell, M. Shepperd, B. Kitchenham, and E. Mendes. 2010. How Reliable Are Systematic Reviews in Empirical Software Engineering?," *IEEE Transactions on Software Engineering*, 36 (5), 676–687.

[4] B. A. Kitchenham, D. Budgen, and P. Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press.

[5] H. Zhang and M. Ali Babar. 2010. On Searching Relevant Studies in Software Engineering. In *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, Swindon, UK, 111–120.

[6] pv-singh/DLs-for-SLRs, *GitHub*. [Online]. Available: https://github.com/pv-singh/DLs-for-SLRs. [Accessed: 15-Feb-2017].

[7] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang. 2016. A Map of Threats to Validity of Systematic Literature Reviews in Software Engineering. In *Proc. of Asia-Pacific Software Engineering Conference*. IEEE, South Korea, 153-160.

[8] B. A. Kitchenham. 1996. Evaluating Software Engineering Methods and Tool Part 1: The Evaluation Context and Evaluation Methods. *SIGSOFT Softw. Eng. Notes*, 21, 1 (1996), 11-14.

[9] B. A. Kitchenham and L. Jones. 1997. Evaluating Software Engineering Methods and Tools Part 6: Identifying and Scoring Features. *SIGSOFT Softw. Eng. Notes*, 22, 2 (1997), 6-18.

[10] B. A. Kitchenham and L. Jones. 1997. Evaluating SW Eng. Methods and Tools, Part 8: Analysing a Feature Analysis Evaluation. *SIGSOFT Softw. Eng. Notes*, 22, 5 (1997), 10-12.

[11] C. Marshall, P. Brereton, and B. Kitchenham. 2014. Tools to Support Systematic Reviews in Software Engineering: A Feature Analysis. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA, 13:1-13:10.

[12] P. Rodríguez *et al.* 2017. Continuous deployment of software intensive products and services: A systematic mapping study. *J. Sys. & Soft.*, 123 (2017), 263-291.

[13] A. Adewumi, S. Misra, N. Omoregbe, B. Crawford, and R. Soto. 2016. A systematic literature review of open source software quality assessment models. *Springerplus*, 5, 1 (2016).

[14] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, and M. Mejías. 2016. Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review. *Information and Software Technology*, 71 (2016), 92-107.

[15] B. Kitchenham. 2010. What's Up with Software Metrics? - A Preliminary Mapping Study. *Journal of System and Software*, 83, 1 (2010), 37-51.

[16] L. Chen, M. A. Babar, and H. Zhang. 2010. Towards an Evidence-based Understanding of Electronic Data Sources, In *Proc. of the Int'l Conference on Evaluation and Assessment in Software Engineering*, Swindon, UK, 135-138.

[17] J. Bailey, C. Zhang, D. Budgen, M. Turner, and S. Charters. 2007. Search Engine Overlaps: Do They Agree or Disagree?. In *Proc. of the Second Int'l Workshop on Realising Evidence-Based Soft. Engg.*, Washington, DC, USA.

[18] M. Ghafari, M. Saleh, and T. Ebrahimi. 2012. A Federated Search Approach to Facilitate Systematic Literature Review in Software Engineering. *International Journal of Software Engineering and Applications*, 3, 2 (2012), 13-24.

[19] O. Dieste, A. Grimán, and N. Juristo. 2009. Developing search strategies for detecting relevant experiments, *Emp. Soft. Engg.*, 14, 5 (2009), 513-539.

[20] M. Sampson, J. McGowan, E. Cogo, J. Grimshaw, D. Moher, and C. Lefebvre. 2009. An evidence-based practice guideline for the peer review of electronic search strategies. *Journal of Clinical Epidemiology*, 62, 9 (2009), 944-952.