# A Systematic Mapping Study of Information Retrieval Approaches Applied to Requirements Trace Recovery

Bangchao Wang[1,2], Heng Wang[2], Ruiqi Luo[1,2*], Sen Zhang[2], Qiang Zhu[1,2],
[1] Engineering Research Center of Hubei Province for Clothing Information, Wuhan Textile University, Wuhan, China
[2] School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China

***Abstract*— Context: Requirements trace recovery (RTR) is always time-consuming, tedious, and fallible. There has been a growing interest in applying information retrieval (IR) to automate the process of recover trace links between requirements artifacts and other software artifacts. Objective: In this review, our objective is to identify the state-of-the-art of how IR has been explored to automate RTR and provide an overview of the research at the intersection of these two fields. Method: A systematic mapping study has been conducted, searching the main scientific databases. The search retrieved 1587 citations and 34 articles are retained as primary studies. Results: The results show the most active authors and publication distribution. It presents four kinds of IR models and 21 enhancement strategies applied to perform RTR. Besides, the lists of 37 experimental datasets and 9 measures, commonly used together to evaluate IR-based RTR approaches, are provided. Conclusions: Vector Space Model (VSM) and Latent Semantic Index (LSI) are the most two studied IR models used in RTR. CoEST becomes the most popular, convenient and stable source of datasets. Precision and Recall are the most common measures used to evaluate the performance of IR methods. Overall, IR-based RTR is becoming an increasingly mature cross research field.**

***Keywords*— *requirements trace recovery, information retrieval, systematic mapping study***

## I. Introduction

Requirements Traceability (RT) is defined as ‘the ability to describe and follow the life of a requirement in both a forward and backward direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)’ [1]. Requirements trace recovery (RTR), as an important activity in RT, can help software engineers discover dependencies that exist between requirements artifacts and other software artifacts, evaluate the requirements coverage rate and calculate the influence of requirements change etc. [2].

In recent years, most extensive efforts have been devoted to studying information retrieval (IR) based RTR approaches in RT research community [2]. By introducing IR, it greatly alleviates the problems of heavy manual workload, difficult maintenance, and error-prone problems faced by traditional approaches [2].

This systematic mapping study (SMS) provides evidence-based insights that can help researchers gain a good understanding of IR-based RTR approaches. We believe that readers interested in RT can use this paper as a map for finding studies relevant to their situation.

The remainder of this paper is organized as follows. Section II analyzes the related works, Section III reports the details of our research methodology and logistics, Section IV provides the main findings from our SMS, and Section V discusses the validity threats. Finally, Section VI concludes this study.

## II. Related Works

We found only two systematic reviews related to IR-based RTR [3][4]. The one by Saleem M et al. [3] focus on surveying whether term mismatch is a real barrier for IR-based RT approaches. Besides, this review summarizes the approaches that attempt to solve the term mismatch problem. Since these summarized approaches are only a part of the whole IR-based traceability approaches. Therefore, this review cannot provide an overview of the research at the intersection of IR and RTR.

The other one by Borg M et al. [4] surveys the state-of-the-art of IR-based software traceability (ST). Since ST has a wider research scope than RT, some IR-based RTR approaches are presented in this review. However, the time interval of the primary studies in [4] is before 2012, which indicates that more recent progress in IR-based RTR approaches has not yet been summarised.

Overall, these works do not provide the recent overview of the research at IR-based RTR. Therefore, to the best of our knowledge, there is no SMS on the status of the IR-based RTR. Our work covers this gap.

In this review, our objective is to identify the state-of-the-art of how IR has been explored to automate RTR and provide an overview of the research at the intersection of these two fields.

## III. Research Method

We have conducted an SMS (which is a well-defined and rigorous method) to identify and interpret relevant studies regarding a particular research question, topic area, or phenomenon of interest [5]. The goal of an SMS is to provide a fair, credible, and unbiased evaluation of a research topic using a trustworthy, rigorous, and auditable method. Hence, SMS is an appropriate method for our research, which is aimed at identifying the overall status of IR-based RTR approaches.

### *A. Research questions*

Many relevant studies on IR-based RTR approaches appeared during 2012–2021, and recent progress has not been summarised. To identify the overall status of IR-based RTR, we defined the following research questions (RQs):

RQ1. What are the authors/venues of the primary studies?

RQ2. Which IR models and enhancement strategies have been applied to perform RTR?

RQ3. Which datasets have been applied to verify IR-based RTR approaches?

RQ4. Which measures have been applied to evaluate the performance of IR-based RTR approaches?

### *B. Search process*

We design an SMS protocol to guide the search process based on the SMS guidelines [5]. Relevant papers are

* Corresponding author: Ruiqi Luo (Email: rqluo@wtu.edu.cn).
DOI reference number: 10.18293/SEKE2022-098

retrieved automatically from the databases.

**1) Inclusion and exclusion criteria.**

Once the potentially relevant studies have been obtained, their actual relevance needs to be assessed. We defined the following inclusion and exclusion criteria to select studies from the search results based on the SMS guidelines [5].

**Inclusion criteria:**

I1: The time span of the study is 2012.1—2021.12.

I2: The research topic of study must be IR-based RTR.

I3: The study is not a review paper.

I4: The papers are written in English.

I5: When two papers with the same technology and topic are provided by the same author, we select the one that is described in greater detail.

**Exclusion criteria:**

E1: The time span of the study is not during 2012.1—2021.12.

E2: The research topic of study is not IR-based RTR.

E3: The study is a review paper.

E4: The paper is not written in English.

E5: When two papers with the same technology and topic are provided by the same author, we exclude the one that is described less thoroughly.

**2) Search scope.**

**Time period.** We specify the time period of the published studies for this SMS from January 2012 to December 2021, which is when we started this SMS.

**Electronic databases.** Based on the suggestion in [5] and the access authority of our institution, the following databases are selected as the primary study sources: IEEE, Google Scholar, Elsevier, EI Compendex, and Springer.

**3) Search terms.**

We use population, intervention, comparison, and outcome (PICO) criteria to define the search terms for database search based on the SMS guidelines [5]. Table I shows the search terms in population and intervention.

TABLE I. LIST OF SEARCH TERMS IN POPULATION, INTERVENTION

| PI | Search terms |
|---|---|
| **Population(P)** | requirement traceability, requirement trace, requirement tracing, requirement traceability recovery |
| **Intervention (I)** | information retrieval, IR, semantic |

**Population:** The population in this SMS is 'Requirements Traceability'. We use words that are synonymous to RT as the population (e.g., 'Requirements Tracing', 'Requirements Trace', and 'Requirements Trace Recovery').

**Intervention:** The intervention is 'information retrieval'. We use the word 'information retrieval', 'IR' and 'semantic' as the intervention.

**Comparison:** Because there is no comparative approach for this review according to the SMS guidelines [5], the part of the comparison specified in PICO is not considered in the construction of search terms.

**Outcome:** Because there is no outcome for this review according to the SMS guidelines [5], the part of the outcome specified in PICO is not considered in the construction of search terms.

**4) Study search and selection.**

Fig. 1 summarizes the study search and selection results. The overall process consists of the following phases.

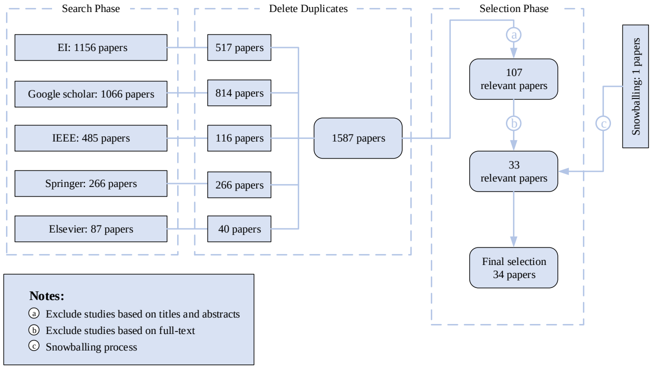


Fig. 1. Study search and selection results.

**Phase 1: Keyword-based literature retrieval**

The search terms (defined in Table I) are applied only to the title, keyword, and abstract because a full text search would yield a large number of irrelevant results.

**Phase 2: 1st round of literature filtering**

The titles, abstracts, and keywords of all potential primary studies are checked by the second and fourth authors against inclusion and exclusion criteria. If it is difficult to determine whether a paper should be included or not, it is reserved for the next phase.

**Phase 3: 2nd round of literature filtering**

In this round, the first and third authors read the full text to determine whether the paper should be included or not based on the inclusion and exclusion criteria. When an agreement cannot be reached, they are asked to state the reasons for inclusion/exclusion to an arbitration panel.

**Phase 4: Snowballing**

After the filtering, 'snowballing' is conducted to find omitted papers. We adopt the snowballing process proposed by Claes Wohlin [6] to iteratively search the reference list and papers cited in a study until no new papers are found. Then, the new papers are checked against the inclusion and exclusion criteria.

## C. *Data extraction and synthesis*

When conducting data extraction, the authors carefully read the primary studies and have conducted a strict peer-review process. Before official extraction, a pilot of the data extraction has been performed. During official extraction, data are extracted based on a detailed set of questions. We kept a record of the extracted information in a spreadsheet for later analysis.

For synthesizing data, qualitative and quantitative data is involved. When synthesizing the data, this process was supported by the extracted data spreadsheet mentioned above.

After performing a separate analysis of the qualitative and quantitative results, we investigate the combination of both sources of evidence. Additionally, we also explore the combination of results from different research questions to build an evidence map that identifies research trends and gaps according to multiple perspectives (questions-answers).

# IV. RESULTS

This section presents the results of the SMS and is arranged in the order of the research questions presented in Section III-A.

*A. RQ1. What are the authors/venues of the primary studies?*


| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Nasir Ali | 5 | Nan Niu | 4 | Anas Mahmoud | 4 |
| Jiale Zhou | 3 | Bangchao Wang | 2 | Rong Peng | 2 |
| Danissa V. Rodriguez | 2 | Doris L. Carver | 2 | Hongyu Kuang | 2 |
| Yue Lu | 2 | Jyoti | 2 | Others(>20) | 1 |

Fig. 2. Authors of the primary studies.

Fig. 2 shows the author distribution. More than 100 authors have appeared in the 34 primary studies. Nasir Ali, Nan Niu, Anas Mahmoud are the top 3 most active authors. It should be noted that some authors that appear only once are not presented in Fig. 2 because of the space limit.


REC **4** ICPC **2** Others(28) **1**

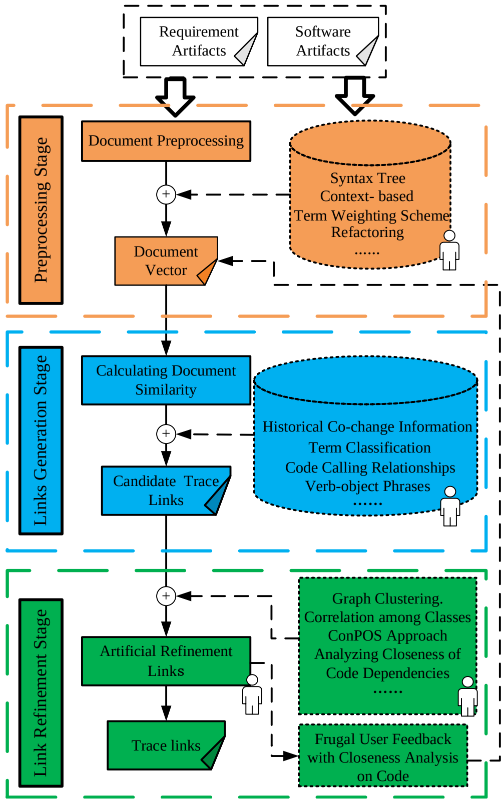
Fig. 3. Conferences and Journals in which the studies are published.

Thirty-four primary studies have been published in 30 types of conferences and journal, as shown in Fig. 3. The RE conference (REC), and ICPC are the top 2 places where the most papers are published.

*B. RQ2. Which IR models and enhancement strategies have been applied to perform RTR?*

The general process of IR-based trace recovery is shown in Fig. 4, which usually includes three stages: **1) Preprocessing Stage:** both the source (Requirements Artifact) and target artifacts (Software Artifact) are regarded as text, and noise information is removed through certain document preprocessing methods to generate document representation that is convenient for subsequent processing; **2) Trace Links Generation (Recovery) stage:** Calculating the similarity between the two artifacts using various document similarity calculation methods, sorting according to the similarity scores, and selecting the candidate trace links according to the set threshold. **3) Trace Link Refinement Stage:** The candidate trace links are refined by manual or semi-automatic methods, and the trace links are finally confirmed by the analyst.

Usually, after preprocessing, trace links can be automatically established using various types of IR-based models. As shown in Table II, Vector Space Model (VSM) and Latent Semantic Index (LSI) are the most two commonly used IR-based models in trace link generation. It should be noted that some studies make the IR model as an integral part of the whole approaches, such as [26][31][33]. Another notable conclusion is that more than half primary studies have presented, used or verified more than one IR models.




**Note: "[person icon]" represents "manual intervention"; "⊕" represents "introducing enhancement strategies".**

Fig. 4. The general process of IR-based trace recovery

To improve the performance, various types of enhancement strategies are proposed. For example, as shown in Fig. 4, Syntax Tree [27] and Refactoring [15][32][36] are used to reduce the adverse effects caused by inconsistent terminology or missing, misplaced signs in textual artifacts. The details of enhancement strategies for IR-based RTR are listed in Table III. From this table, most strategies focus on how to improve the performance of the VSM and LSI.

TABLE II. IR MODELS USED IN REQUIREMENTS TRACE RECOVERY

| | | | |
|---|---|---|---|
| IR Models | Algebraic Models | VSM | [7] [8] [10] [11] [13] [14] [15] [18] [19] [20] [21] [22] [23] [24] [25] [27] [28] [29] [32] [34] [35] [36] [37] [38] [39] [40] |
| | | LSI | [9] [11] [12] [14] [15] [16] [17] [23] [30] [32] [35] [36] [38] [40] |
| | Probabilistic Models | JS | [11] [22] [24] [39] [40] |
| | | TM | [21] [29] |
| | Other | [26] [31] [33] | |

**Note: VSM: Vector Space Model, LSI: Latent Semantic Index, JS: Jensen-Shannon Model, TM: Topic Model.**

*C. RQ3. Which datasets have been applied to verify IR-based RTR approaches?*

The answer for this RQ can be used as a guideline for the researchers to select datasets based on their research needs and the characteristics of the datasets. For instance, for each of these datasets, Table IV provides a link to the open-source datasets, along with other meta-data details associated with it, such as primary studies that used it, trace space (the maximum counts of trace links), and other characteristics of the dataset.

TABLE III. LIST OF ENHANCEMENT STRATEGIES FOR IR-BASED REQUIREMENTS TRACE RECOVERY APPROACHES

| Strategy | IR model | | | | Applying Phrase | Strategy Characteristics |
|---|---|---|---|---|---|---|
| | VSM | LSI | JS | TM | | |
| Context- based [8][10][37] | ● | | | | **P** | Separating intent from context in requirements |
| Improved Term Weighting Scheme [12][16] | | ● | | | **P** | Proposing an improved term weighting scheme, namely, Developers Preferred Term Frequency/Inverse Document Frequency (DPTF/IDF) |
| Refactoring [15][32][36] | ● | ● | | | **P** | Solving the problem of missing symbols, misplaced symbols and repeated symbols |
| Syntax Tree [27] | ● | | | | **P** | Primary identifier keywords are converted to comment keywords by their similarity in appearance in the syntax tree location |
| Verb-object Phrases [7] | ● | | | | **P** | Extracting verb-object phrases as main information and essential meaning |
| Analyzing Close Relations [13] | ● | | | | **G** | Calculating the close relations (semantic similarity) between target artifacts |
| Term Classification [17][30] | | ● | | | **G** | Categorizing class names, comments, and all other terms in code |
| Model-Driven Engineering (MDE) [19] | ● | | | | **G** | Combining use of both MDE and IR, analyzing the textual information (organization and hierarchy) contained in the model to retrieve implicit links between documents |
| Hybrid Method [21][29] | ● | | | ● | **G** | Combing VSM and BTM which can help relieve data sparsity caused by short text |
| Genetic Algorithm [29] | | | | ● | **G** | Configuring initial parameters of BTM by introducing Genetic Algorithm |
| Code Calling Relationships [20] | ● | | | | **G** | Identifying errors between requirements and code traces by code-calling relationships |
| Historical Co-change Information [23] | ● | ● | | | **G** | Taking the processed corpora and co-change information of classes as input to reorder and filter baseline links |
| Dynamic Integration of Structural and Co-change Coupling [28] | ● | | | | **G** | Retrieving indirect links based on weighted integration of structural coupling and class coupling based on change history |
| Configuration Management Log [35][38] | ● | ● | | | **G** | Restoring links by finding revisions in the configuration management log that contain words related to requirements |
| Frugal User Feedback with Closeness Analysis on Code [40] | ● | ● | ● | | **R** | Introducing only a small amount of user feedback into the closeness analysis on call and data dependencies in code |
| User Feedback [35] | ● | | | | **R** | Introducing user validation for candidate links to improve accuracy |
| Analyzing Closeness of Code Dependencies [11] | ● | ● | ● | | **R** | Quantifying the interaction degree of call dependency and data dependency between two code classes |
| Class Clustering [17] | | ● | | | **R** | The products in the clustering have similar trace relationships |
| Correlation among Classes [25] | ● | | | | **R** | Using structural or co-changing dependencies or both to find correlations between classes and use these dependencies to verify traceability links |
| Graph Clustering [34] | ● | | | | **R** | Information about the cohesion of artifacts within a level of refinement helps improve the trace retrieval process between levels of refinement |
| ConPOS approach [39] | ● | | ● | | **R** | Pruning trace links using the primary POS classification and apply constraints to recovery as a filtering process |

**Note: " ● " represents support; "P" represents Preprocessing Stage, "G" represents Links Generation Stage, "R" represents Links Refinement Stage.**

TABLE IV. DATASETS' INFORMATION AND THE STUDIED PAPERS WHICH USED THE DATASETS

| Dataset Name | Source Artifacts (Number) | Target Artifacts (Number) | Space | True Links | Scale | Freq. | Resource links | Reference |
|---|---|---|---|---|---|---|---|---|
| iTrust | Use cases (34) | Code (243) | 8262 | 603 | Large | 16 | http://www.coest.org/ | [8][11][12][13][14][15][16][23][25][26][28][30][32][36][39][40] |
| | Requirements (50) | Code (299) | 14950 | 314 | | | | |
| eTour | Use Cases (58) | Code (116) | 6728 | 308 | Large | 11 | http://www.coest.org/ | [7][8][14][15][18][24][29][30][31][32][36] |
| | Requirements (58) | Code (116) | 6728 | 366 | | | | |
| EasyClinic | Requirements (30) | Code (47) | 1410 | 93 | Small | 6 | http://www.coest.org/ | [7][9][13][21][29][34] |
| | Use Cases (30) | Test Cases (63) | 1890 | 63 | | | | |
| CM-1 | High-level Requirements (235) | Low-level Requirements (220) | 51700 | 4050 | Large | 5 | http://www.coest.org/ | [9][13][14][30][34] |
| | High-level Requirements (235) | Design (220) | 51700 | 361 | | | | |
| | Requirements (235) | Design (220) | 51700 | 361 | | | | |
| | Requirements (235) | Use Case (Unclear) | Unclear | Unclear | Unclear | | | |
| Pooka | Requirement (298) | Code (90) | 26820 | 546 | Large | 5 | http://www.suberic.net/pooka/ | [12][16][22][23][39] |
| EBT | Requirements (41) | Test Cases (25) | 1025 | 51 | Small | 4 | http://www.coest.org/ | [18][21][24][29] |
| | requirements (40) | Code (50) | 2000 | 98 | | | | |
| GanttProject | Requirements (16) | Code (124) | 1173 | 315 | Small | 4 | http://www.ganttproject.biz | [11][13][20][40] |
| Albergate | requirements (17) | Code (55) | 935 | 54 | Small | 3 | http://www.coest.org/ | [18][24][31] |
| SIP Communicator | Requirements (82) | Code (1771) | 145222 | 871 | Large | 3 | http://www.jitsi.org | [22][23][39] |
| WARC | Non-functional Requirements (21) | Software Requirements Specification (89) | 1869 | 58 | Small | 2 | http://www.coest.org/ | [21][29] |
| | Functional Requirements (43) | Software Requirements Specification (89) | 3827 | 78 | Large | | | |
| GANNT | High-level Requirements (17) | Low-level Requirements (69) | 1173 | 68 | Small | 1 | http://www.coest.org/ | [13] |
| jEdit v4.3 | Requirements (34) | Code (483) | 16422 | Unclear | Large | 1 | http://www.jedit.org. | [22] |
| Infinispan | Requirements (237) | Code (388) | 91956 | 1515 | Large | 1 | http://infinispan.org/ | [40] |
| Lucene | Requirements (116) | Code (413) | 47908 | 744 | Large | 1 | http://lucene.apache.org | [12] |
| Rhino v1.6 | Requirements (268) | Code (138) | 36984 | Unclear | Large | 1 | http://www.mozilla.org/rhino/ | [22] |
| Lynx | Requirements (90) | Code (298) | 26820 | 507 | Large | 1 | http://lynx.isc.org/ | [39] |
| Maven | Requirements (36) | Code (94) | 3384 | 155 | Large | 1 | http://maven.apache.org/ | [40] |
| Pig | Requirements (68) | Code (236) | 16048 | 356 | Large | 1 | https://pig.apache.org/ | [40] |
| Mylyn | Requirements (Unclear) | Code (Unclear) | Unclear | Unclear | Unclear | 1 | http://www.eclipse.org/mylyn/developers | [26] |

Note: There are 18 datasets that cannot be obtained, i.e., Chess [20], CUnit [38], iBooks [7], iRobot [10], iTruck [10], iSudoku [10], jHotDraw (JHD) [11][20], SMS [7], MODIS [9], MR0 [9], MR1 [9], MR2 [9], Pine [13], VideoOnDemand (VoD) [20], WDS [30][36], Waterloo [34], LEDA [31], network control system [38].

Thirty-seven datasets used for experimental validation were extracted from thirty-four primary studies, as shown in Table IV. The iTrust, eTour and EasyClinic are the top 3 most popular datasets, which all are provided by the Center of Excellence for Software & Systems Traceability (CoEST). Besides, In the nineteen open-source datasets, eight datasets are provided by CoEST, and the resource can be found at http://www.coest.org/. These nineteen open-source datasets have been used sixty-eight times. More than 70% (48/68, 70.6%) of cases, the used datasets are from CoEST. Obviously, it becomes the most popular, convenient and stable source of datasets.

On the other side, twelve primary studies (12/34, 35.3%) use the eighteen non-open-source datasets, as shown at the bottom of Table IV. It prevents researchers from reproducing experiments and using datasets.

TABLE V. NUMBER OF USED DATASETS DISTRIBUTION FOR PRIMARY STUDIES

| Number of datasets used in primary studies | Number of primary studies | Proportion |
|---|---|---|
| 0 | 2 | 5.88% |
| 1 | 6 | 17.65% |
| 2 | 5 | 14.71% |
| 3 | 12 | 35.29% |
| 4 | 5 | 14.71% |
| 5 | 3 | 8.82% |
| >5 | 1 | 2.94% |
| Total | 34 | 100% |

As shown in Table V, the fact that nearly a quarter (8/34, 23.5%) of primary studies validated their methods by applying to less than two datasets is a threat to the external validity [6]. The more datasets used, the more general the method becomes. To mitigate external threats, researchers are encouraged to use various types of datasets as many as possible.

*RQ4. Which measures have been applied to evaluate the performance of IR-based RTR approaches?*

After trace links are generated, the evaluation of them is an indispensable task. In most cases, precision and recall are the most common measures used to evaluate the performance of IR-based RTR approaches, as shown in Table VI. Once further evaluation is needed, the secondary measures, such as MAP, AP, DiffAR, Lag, Selectivity and Cliff's Delta, are good candidates. Due to the space limit, no specific details to the performance measures, which are used to evaluated IR-based RTR approaches, are given here.

TABLE VI. PERFORMANCE MEASURES USED IN PRIMARY STUDIES

| Categories | Measures | Primary Studies |
|---|---|---|
| Primary Measure | Recall | [7][10][11][12][13][16][17][18][20][21] [22][23][24][25][26][27][28][29][30][31] [32][34] [35][36][37][38][39][40] |
| | Precision | |
| | F-Measure | [7][12][22][25][27][28][29][35][38] |
| Secondary Measure | MAP | [8][9][10][11][13][14][32][33][34][36] [39][40] |
| | AP | [11][13][33][34][39][40] |
| | DiffAR | [14][32][36] |
| | Lag | [14] |
| | Selectivity | [29] |
| | Cliff's Delta | [40] |

## V. VALIDITY THREATS DISCUSSION

In this section, we aim to discuss these potential threats that influence the data extraction and the findings of this SMS. According to the guidelines for analyzing the validity threats to SE methods and processes [6], conclusion validity, construct validity, internal validity, external validity will be discussed in the following.

**Conclusion validity:** It is possible that some papers excluded by this review should have been included. To mitigate this type of threat, the selection process and the inclusion and exclusion criteria described in Section III-B are carefully designed and discussed by authors to minimize the risk of exclusion of relevant studies.

**Construct validity:** The main constructs in this SMS are 'requirements traceability' and 'information retrieval'. We respectively use these two terms and their synonyms to ensure that all selected primary studies are relevant to the intersection of these two fields. Meanwhile, snowballing from literature sources is performed complementary to database search to ensure that relevant studies are covered as much as possible.

**Internal validity:** In this SMS, a different participant may end up with different data extraction and analysis results. This may be a threat to the internal validity. To mitigate the threat, the data extraction is performed collaboratively by two authors. Moreover, any conflicts are discussed and resolved by all the authors in this process.

**External validity:** It is concerned with establishing the generalizability of the SMS results, which is related to the degree to which the primary studies are representative of the review topics. To mitigate external threats, the search process described in Section III-B is defined after several trial searches. Moreover, we have tested the coverage and representativeness of retrieved papers.

## VI. CONCLUSIONS

RTR is always heavy manual workload, time-consuming, tedious, and fallible [2]. To alleviate these problems, there has been a growing interest in applying IR to automate the process of recover trace links between requirements artifacts and other software artifacts [2]. In this SMS, we survey the state-of-the-art of how IR has been explored to automate RTR and provide an overview of the research at the intersection of these two fields. By analyzing the 34 primary studies, the following conclusions have been obtained:

Firstly, VSM, LSI, JS and TM are the main kinds of IR models applied to perform RTR during the decade. Besides, 21 enhancement strategies are developed to improve the performance of these models. Researchers are encouraged to use multiple strategies to construct a combination approach.

Secondly, CoEST is proved to be the most popular, convenient and stable source of datasets. Researchers are encouraged to use various types of open-source datasets as many as possible. It helps other researchers to reproduce experiments and validate the proposed approaches.

Thirdly, Precision and Recall are the most common measures used to evaluate the performance of IR methods. Also, MAP is the most popular secondary measures. Researchers are encouraged to evaluate IR methods from different dimensions by applying different measures.

## ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 62102291.

## REFERENCES

[1] O. Gotel and C. W. Finkelstein, "An analysis of the requirements traceability problem," Proceedings of IEEE International Conference on Requirements Engineering,1994, pp. 94-101.

[2] B. Wang, R. Peng, Y. Li, et al.,"Requirements traceability technologies and technology transfer decision support: A systematic review," Journal of Systems and Software, 2018, 146: 59-79.

[3] M. Saleem,and N. M. Minhas ,"Information retrieval based requirement traceability recovery approaches-a systematic literature review," University of Sindh Journal of Information and Communication Technology, 2018, 2(4): 180-188.

[4] M. Borg, P. Runeson and A. Ardoe, "Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability,"Empirical Software Engineering, 2014.

[5] P. Kai, S. Vakkalanka and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update, " Information and Software Technology, 2015, 64:1-18.

[6] C. Wohlin, P. Runeson, M. Host, et al.,"Experimentation in Software Engineering," Springer-Verlag, Berlin, Heidelberg, 2012.

[7] Y. Zhang, C. Wan and B. Jin, "An empirical study on recovering requirement-to-code links," 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),pp.121-126.

[8] J. Zhou, Y. Lu and K. Lundqvist, "An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis," 2013

[9] S. Eder, H. Femmer, B. Hauptmann and M. Junker, "Configuring Latent Semantic Indexing for Requirements Tracing," 2015 IEEE/ACM 2nd International Workshop on Requirements Engineering and Testing, 2015, pp. 27-33.

[10] J. Zhou, Y. Lu and K. Lundqvist, "A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems," 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, 2013, pp. 252-259.

[11] H. Kuang, J. Nie, H. Hu, et al., "Analyzing closeness of code dependencies for improving IR-based Traceability Recovery". 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2017, pp. 68-78.

[12] N. Ali, Z. Sharafi, Y. G. Gueheneuc, et al., "An empirical study on the importance of source code entities for requirements traceability," Empirical Software Engineering, 2015 ,20(2):442-478.

[13] H. Wang, G. Shen, Z. Huang, et al.,"Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery," Frontiers of Information Technology & Electronic Engineering, 2021,22(7):957-968.

[14] A. Mahmoud, N. Niu and S. Xu, "A semantic relatedness approach for traceability link recovery," 2012 20th IEEE International Conference on Program Comprehension (ICPC), 2012, pp. 183-192.

[15] F. Faiz, R. Easmin and A. U. Gias, "Achieving Better Requirements to Code Traceability: Which Refactoring Should Be Done First?," 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC), 2016, pp. 9-14.

[16] N. Ali, Z. Sharafl, Y. -G. Guéhéneuc and G. Antoniol, "An empirical study on requirements traceability using eye-tracking," 2012 28th IEEE International Conference on Software Maintenance (ICSM), 2012, pp. 191-200.

[17] J. Shao, W. Wu and P. Geng, "An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing," 13th International Conference on Computational Science & Its Applications, Springer Berlin Heidelberg, 2013.

[18] D. V. Rodriguez and D. L. Carver, "An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery," 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 2020, pp. 1145-1153.

[19] N. Sannier and B. Baudry, "Toward multilevel textual requirements traceability using model-driven engineering and information retrieval," 2012 Second IEEE International Workshop on Model-Driven Requirements Engineering (MoDRE), 2012, pp. 29-38.

[20] A. Ghabi and A. Egyed, "Code patterns for automatically validating requirements-to-code traces," 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, 2012, pp. 200-209.

[21] B. Wang, R. Peng, Z. Wang, et al., "An Automated Hybrid Approach for Generating Requirements Trace Links," 31st International Conference on Software Engineering and Knowledge Engineering, SEKE 2019.

[22] N. Ali, Guéhéneuc, Yann-Gal, and G. Antoniol, "Trustrace: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links," IEEE Transactions on Software Engineering, 2013, 39(5):725-741.

[23] N. Ali, F. Jaafar and A. E. Hassan, "Leveraging historical co-change information for requirements traceability," 2013 20th Working Conference on Reverse Engineering (WCRE), 2013, pp. 361-370.

[24] D. V. Rodriguez and D. L. Carver, "Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery," 2020 IEEE International Conference on Electro Information Technology (EIT), 2020, pp. 271-280.

[25] Jyoti and J. K. Chhabra, "Filtering of false positives from IR-based traceability links among software artifacts," 2017 2nd International Conference for Convergence in Technology (I2CT), 2017, pp. 1111-1115.

[26] P Hu¨Bner. "Quality Improvements for Trace Links between Source Code and Requirements," REFSQ-2016 Workshops, co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2016.

[27] S. Nagano, Y. Ichikawa and T. Kobayashi, "Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts," 2012 IEEE 36th Annual Computer Software and Applications Conference, 2012, pp. 11-18.

[28] Jyoti and J. K. Chhabra, "Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling," International Conference on Advanced Informatics for Computing Research. Springer, Singapore, 2017.

[29] B. Wang, R. Peng, Z. Wang, et al., “An Automated Hybrid Approach for Generating Requirements Trace Links,” International Journal of Software Engineering and Knowledge Engineering, 2020.

[30] N. Niu and A. Mahmoud, "Enhancing candidate link generation for requirements tracing: The cluster hypothesis revisited," 2012 20th IEEE International Requirements Engineering Conference (RE), 2012, pp. 81-90.

[31] A. Ghannem, M. S. Hamdi, M. Kessentini and H. H. Ammar, "Search-based requirements traceability recovery: A multi-objective approach," 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1183-1190.

[32] A. Mahmoud and N. Niu, "Supporting requirements traceability through refactoring," 2013 21st IEEE International Requirements Engineering Conference (RE), 2013, pp. 32-41.

[33] R. Jain, S. Ghaisas and A. Sureka, “SANAYOJAN: a framework for traceability link recovery between use-cases in software requirement specification and regulatory documents,” 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, RAISE 2014.

[34] P. Rempel, P. Mäder and T. Kuschke, "Towards feature-aware retrieval of refinement traces," 2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE), 2013, pp. 100-104.

[35] R. Tsuchiya, H. Washizaki, Y. Fukazawa, et al., "Interactive Recovery of Requirements Traceability Links Using User Feedback and Configuration Management Logs," 27th International Conference on Advanced Information Systems Engineering, CAISE 2015, Springer International Publishing, 2015.

[36] A. Mahmoud and N. Niu, "Supporting requirements traceability through refactoring," 2013 21st IEEE International Requirements Engineering Conference (RE), 2013, pp.32-41.

[37] J. Zhou, "Requirements development and management of embedded real-time systems," 2014 IEEE 22nd International Requirements Engineering Conference(RE), 2014, PP. 479-484.

[38] R. Tsuchiya, H. Washizaki, Y. Fukazawa, et al., "Recovering Traceability Links between Requirements and Source Code Using the Configuration Management Log," IEICETransactions on Information and Systems, 2015, 98-D(4).

[39] N. Ali, H. Cai , A. Hamou-Lhadj , et al., "Exploiting Parts-of-Speech for effective automated requirements traceability," Information and software technology, 2019, pp.126-141.

[40] H. Kuang, H. Gao, H. Hu, et al., "Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery," 2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC), 2019, pp. 369-379.