



Information Retrieval based requirement traceability recovery approaches- A systematic literature review

Muhammad Saleem¹, Nasir Mehmood Minhas²

¹UIIT, PMAS Arid Agriculture University Rawalpindi, Pakistan,

²SERL Sweden, Blekinge Institute of Technology, Karlskrona, Sweden
msaleembhatti@hotmail.com, nasir.mehmood.minhas@bth.se

Abstract: The term traceability is an important concept regarding software development. It enables software engineers to trace requirements from their origin to fulfillment. Maintaining traceability manually is a time consuming and expensive job. Information retrieval methods provide a mean of automation for requirement traceability. A visible number of IR based traceability techniques have been proposed in the literature, but the adoption of these techniques in the industry is limited. In this paper, we examine the information retrieval-based traceability recovery approaches through systematic literature review. We presented a synthesis of these techniques. We also identified challenges that are potentially limiting the adoption of IR based traceability recovery approaches. We conclude that term mismatch is a major barrier faced by IR based approaches. We also did classify the approaches that are attempting to solve the term mismatch problem.

Keywords: Information retrieval, Requirement traceability, Term mismatch, Challenges;

I. INTRODUCTION

Continuously changing requirements causes many problems like scope creep or analyzing impact of change etc. Requirement traceability (RT) helps to manage such problems. RT is defined as the “ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)” [37]. Manual approaches of managing trace links consume a lot of time. To cope this, it is essential to automate this process. Most of the software artifacts are in natural languages. Therefore, recovering trace links between such artifacts can be considered as an information retrieval (IR) problem. Many automated requirement traceability recovery techniques and tools have been using IR approach [1], [23]. IR makes use of textual similarity between documents and query to consider them related. If people use consistent vocabulary during software development, then textually similar artifacts share similar terms and can be traced from each other through IR techniques [15]. Such techniques use high level artifacts like requirement documents, use cases or source code etc. to extract textual similarity between them to establish expected trace links. While IR promised to provide automated solutions of traceability problem, but practitioners are still reluctant to use such tools due

to lack of precise results. This study summarizes the IR based requirement traceability recovery approaches to identify the major faced by them.

II. RESEARCH METHODOLOGY

The purpose of this study is to summarize the existing knowledge and identify the gaps to suggest future directions in the area. We conduct a systematic literature review (SLR) following the guidelines of [45], [44]. SLR provides a procedure to identify, evaluate and understand the research questions (RQ) by means of all available publications and research in that area. This systematic review consists of three phases: Planning, executing and reporting. In the planning phase a review protocol is developed. Executing phase includes execution of review protocol like identification, selection of studies and their quality assessment. The reporting phase refers to reporting and analyzing the results [45], [44].

This SLR aims to synthesize the work regarding the term mismatch issue in IR based requirement traceability recovery approaches.

A. Review Protocol Development

1) Need for study: Requirement traceability (RT) is an essential aspect of software development. Practitioners require the precise generation of trace links before they integrate automated trace retrieval methods into their software development processes [44]. We conducted a pilot search to identify secondary studies in the area. We perform search on IEEE Xplore, ACM digital library, SpringerLink, ScienceDirect, Elsevier. We used general terms like information retrieval, traceability, reviews, mappings, as our

search string. We did not find any secondary study in our area of interest. This was the motivation for conducting this systematic literature review. The purpose of this study was to review the status of research in automated IR based traceability recovery approaches focusing on the term mismatch problem.

2) Research questions: The objective of this study is to investigate the following research questions.

RQ1: Does the term mismatch problem is a major barrier to IR based traceability recovery approaches? The objective of this research question is to validate, if term mismatch is a real barrier for IR based traceability approaches.

RQ2: Which are the approaches, that attempt to solve the term mismatch problem?

The objective of this research question is to synthesize the techniques that attempts to solve the term mismatch problem.

RQ3: What are the limitations and advantages of these approaches?

Approaches have some built-in limitations and advantages. Identifying such challenges is very important to solve the term mismatch problem. Identification of challenges is a way forward for the future research.

3) Search strategy: The following search strategy is used for construction of search terms. We take out the key terms from our first research question. Then we defined synonyms for the keywords used in the research question as shown in Table I. In the third step we built the search string by joining the synonyms with operator OR and each keyword using the operator AND. Keywords: barrier, IR, traceability recovery Finally we built our search string as follows.

TABLE I: TERMS AND SYNONYMS USED FOR SEARCH STRING

Terms	Synonyms
Barrier	problems OR issue OR limitation OR Challenge
IR	Information retrieval
Traceability	trace links recovery

Following online resources were explored with the search string to find out relevant studies:

- 1) IEEE Xplore
- 2) ACM digital library
- 3) SpringerLink
- 4) ScienceDirect
- 5) Elsevier

Each research paper was reviewed carefully and the papers that addressed the barriers of IR based traceability recovery approaches was relevant. Then we apply the inclusion and exclusion criteria on the relevant papers while searching on different digital libraries.

4) Study selection process: For the final selection of the primary studies we apply the following inclusion exclusion criteria (IE criteria).

a) Inclusion criteria: Following questions helped us to decide about the inclusion of a study.

- 1) If the full-text of the study available?
- 2) If the study is a journal article, conference or workshop paper.
- 3) If the study claims about barriers of IR based approaches.
- 4) If the study, try to provide some solution of term mismatch problem in IR based automated traceability recovery process.

b) Exclusion Criteria: Following studies will be excluded.

- 1) If the study focusses on other aspects of traceability rather than trace links generation.
- 2) Studies with multiple published instances are included only once, the latest version of the study will be used.
- 3) Unpublished work will not be included.
- 4) Reviews or secondary studies will not be included.

We include the articles based on the title, abstract and conclusion of the research paper by completing following the above-mentioned inclusion criteria if the article completely follows the inclusion criteria it must be selected for quality assessment. Figure 1 show the study selection process.



Figure 1: Study Selection Process

5) Study quality assessment: The selected studies were evaluated for quality following IMRAD structure. A study is of good quality if.

I. Introduction: The study discussed barriers faced by IR approaches.

M. Methodology: Research methodology of the study is clearly defined.

R. Results: Results are presented properly. Internal and external validity threats are mentioned.

A. Analysis: Study compares its results with some base technique on an industrial dataset.

D. Discussion. Limitations of the study were reported. Future directions are given.

B. Review protocol evaluation

As review protocol is very important and is a key part of systematic review. Therefore, it is suggested to evaluate review

protocol before executing it [?]. Therefore, after writing our review protocol it was evaluated by an independent researcher. He suggested to evaluate the quality of studies based on Introduction (I), Methodology (M), Results (R), Analysis (A), Discussion (D), called IMRAD.

C. Execution of review

We explore specific online resources to search for primary studies initially we retrieve 1522 studies based on our search string, after title and abstract screening were 321 studies left. In second step we reject 228 studies based on their conclusion. We select 93 primary studies in first step. Then in second step we applied inclusion/exclusion criteria, 26 more studies were rejected based on our inclusion/exclusion criteria. Then we applied study quality assessment criteria. All journal papers were of good quality. Most of the conference papers were from IEEE transactions and were of good quality. Only 1 conference paper was rejected. Workshop papers were under severe screening, 9 out of 22 workshop papers were not fulfilling our quality assessment criteria therefore rejected. Finally, we selected 57 studies (Table: II) to be included in our review. Figure 2 presents year wise distribution and Figure 3 presents journal, conference or workshop wise distribution of primary studies to be included in our review.

TABLE II: PRIMARY STUDY SELECTION

Library	Total	Title & Abstract Screening	Full Text	IE Criteria	IMRAD criteria
IEEE	535	116	45	35	28
ACM Digital library	320	74	24	13	12
Springer	187	36	9	5	5
Elsevier	190	30	10	5	5
Science Direct	160	28	8	6	4
Other	130	25	7	3	3
Total	1522	321	93	67	57



Figure 2: Year wise distribution of primary studies

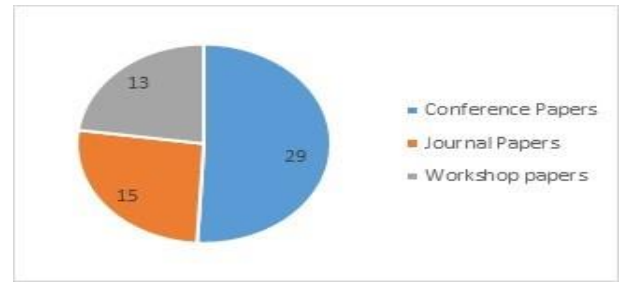


Figure 3: Distribution of primary studies with respect to publication

III. RESULTS

RQ1: Does the term mismatch problem is a major barrier to IR based traceability recovery approaches?

People usually have background knowledge and belong to different domains. A study [31] showed that different people used different names for the same thing almost 80% of the times. During software evolution, new terminology gradually entered into the system that may be inconsistent with the existing terminology or application domain [71], [29], [2]. This difference in vocabulary of the artifacts is called term mismatch problem. From the preliminary review, we found that term mismatch is a primary cause of failure regarding IR based traceability recovery techniques. Therefore, in RQ1 we focused on confirming that term mismatch is a real problem in IR based techniques. We selected a total of 57 primary studies that are utilizing IR based approaches for the automated traceability recovery approaches. Out of 57, authors of 23 studies (40%) highlighted the term mismatch issue as the critical barrier in IR based traceability recovery approaches. We believe that this is a reasonable representation of the population to confirm the fact. Table III shows the list of studies which specifically declared that term mismatch problem as the greatest barrier in IR based traceability recovery techniques.

TABLE III: STUDIES THAT HIGHLIGHTED TERM MISMATCH AS A GREATEST BARRIER

Fact	Primary Studies
Primary studies that specifically declared term mismatch as greatest barrier.	[38], [40], [56], [50], [67], [70], [26], [39], [61], [17], [53], [54], [55], [35], [57], [47], [10], [42], [64], [27], [5], [69], [72], [65]

RQ2: Which are the approaches, that attempt to solve the term mismatch problem?

In this section we will discuss different approaches used for tackling term mismatch problem in primary studies. Nouns can play important role in finding trace links between source and target artifacts. Removing unnecessary terms (noise) from source and target artifacts can improve the accuracy of trace links, Capobianco, 2009 and 2013 [10], [12] authors considered only nouns during traceability recovery process to considerably improve the accuracy. Some other studies [32], [72], [20], [19]

used stemming as a preprocessing step to reduce different morphological forms of terms to a single term to solve term mismatch problem to some extent. In [53], [28], [52] authors suggest solving the term mismatch problem by restoring textual information or removing inconsistent vocabulary from artifacts, this process is called refactoring [30]. Refactoring had a significant positive impact on the precision of extracted trace links, but it was a manual activity. These studies used only term similarity while computing the similarity level between source and target artifacts. Related artifacts may share few terms therefore only term similarity could not recover precise trace links. Antoniol, 2001 [4] and others [57]. Antoniol, 2002 in another study [3], indirectly recovered traceability links between requirement documents and source code by combining textual similarity and structural information like inheritance properties, function calls etc. of source code. Two terms may be related if they co-exist in text, in studies [42], [11], [68] authors used only similar terms to trace source and target artifacts. They used relationships between terms to mitigate the polysemy and synonymy problem, although accuracy of trace links did not significantly improve. Some approaches tried to solve the mismatching terms issue with the help of some non-semantic information [26], [39]. Developers usually work on similar and related tasks, this information called author's context can be used to aid the traceability recovery process. They provide a four-step approach to establish traceability links between requirements and source code. Computing textual similarity, identifying author of each code, defining authors context, integrate code ownership with textual information to produce traceability matrix. Bacchelli, 2010 [6] used emails instead of authors context for similar purpose and [7] used eye-gaze information to identify authors context. In [67], [61], [55], [59], [22], [60], [34] authors proposed to use user feedbacks provided by the software engineer while classifying candidate trace links. The software engineer classifies the link either as correct link or not. Then new links can be obtained based on this feedback and structural information of code. The ranked list is reordered based on this information. Link count information [9] was used to identify scarcely traced artifacts probably due to mismatching terms. Comparing two texts based on phrases rather than individual terms matching can provide more precise results. In [46] authors favor the phrase matching rather than single term matching with explicit goal of improving accuracy. Latent semantic indexing (LSI) is a very important IR model. LSI used a latent structure hidden in text [49] to find trace links between source and target artifacts. Query expansion techniques are most widely used to solve the term mismatch problem. External knowledge bases (KB) can be used to extract synonyms or measure similarity (0 to 1) between the mismatching terms. In [56], [54], [35], [51] authors used normalized google distance Wiki similarity to compute semantic similarity between terms. They observed improvement in precision and recall of traced links. In another study DBpedia was used to extract synonyms of terms to compute similarity level between artifacts [50].

Google, Yahoo, Bing also used as external knowledge base to expand the query artifact with semantically related terms to minimize the probability of term mismatch [36], [16]. External documentation like user stories and descriptions can also be used to expand the source artifact to trace links with a slightly higher precision [8], [18]. Query expansion techniques significantly improved recall as there will be more matching terms. Natural language processing techniques can be applied to get semantic understanding of sentences. In [14] authors applied GATE framework to apply NLP techniques to produce a tree structure of subject, object and predicate in sentences (triplet). Based on this triplet they use DBpedia to extract categories of each triplet to expand the query artifact. Spanoudakis, 2004 [64] used grammatical tagging and rules to generate trace links between source and target artifacts. Machine learning can also be used to expand the query with related terms learned through a training data set (regulation-to-requirements trace links. In [38], [40], [27], [25] authors trained a classifier for addressing the term mismatch problem and improving the quality of trace links. Thesaurus can be used to control the vocabulary of software artifacts. In [66], [24] author used thesaurus as a tool to avoid mismatching term hence avoiding the mismatching terms problem. Some authors also used hybrid approaches [34], [33] to increase the precision of recovered trace links, like term similarity with user feedbacks. In [69], [70] used ontology with IR as hybrid approach by building ontology using IR. A possible classification of these approaches is shown in Figure 3. Distribution of these studies based on classification is shown in Figure 4. It was found that most of the studies used EASY CLINIC data set [61], [10], [5], [12], [20], [19], [59], [22], [9], [18], [11], [49] for results evaluation, which is freely available and is relatively a small dataset. E-Tour is another mostly used dataset [26], [53], [54], [55], [12], [20], [52], [18]. Other datasets were found to be relatively less used like CM-I [54], [55], [12], [46], [49] and MODIS [12], [20], [59], SMOS [26], [61], [8], [18]. Figure 6 demonstrates dataset usage in the research area. Figure 7 displays frequency of different artifacts used by researchers to demonstrate traceability method.

RQ3: What are the advantages and disadvantages of these approaches?

In this section we discussed advantages and limitations of major approaches used to solve term mismatch problem during IR based traceability recovery process. To comprehend the complexity of the problem we take advantage of study [32] which classify term mismatches into three types. Morphological variation: Same term can occur in different forms for example computer, computing or computation are the different forms of the word compute. Lexicon Semantic variation: Different words have same meaning. Like calculate or compute have same meaning (synonymy) or same words have different meaning in different contexts (polysemy). Lexicon Semantic variation: Different words have same meaning. Like calculate or compute

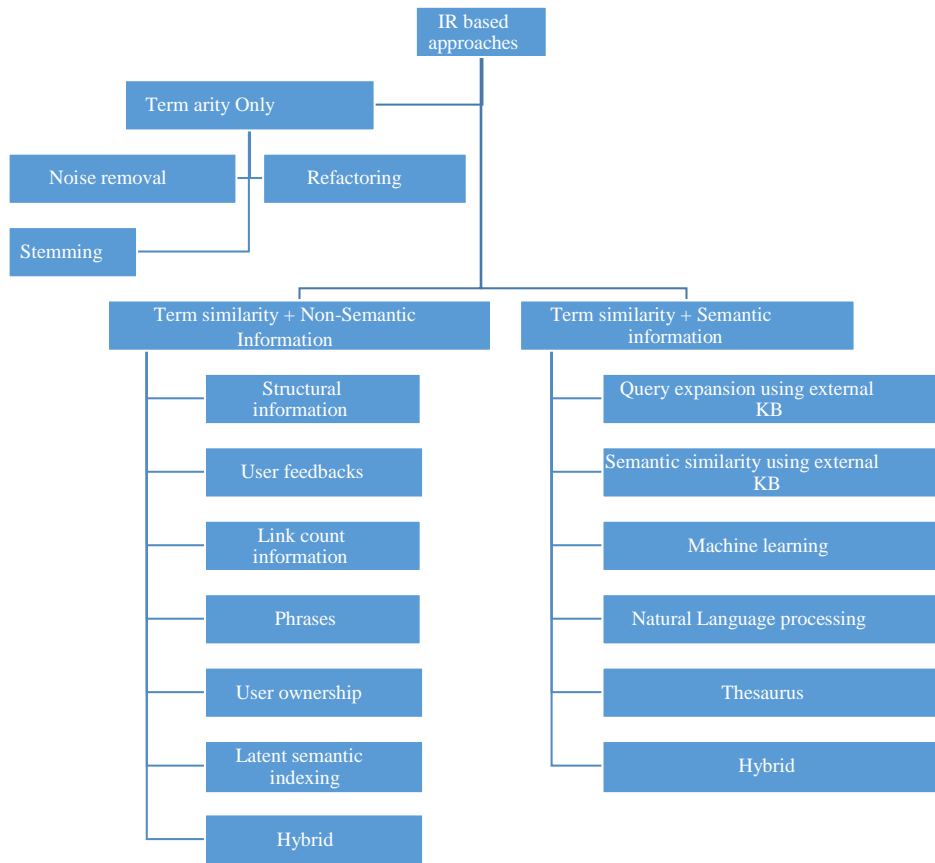


Figure 4: Classification of IR based traceability recovery techniques

have same meaning (synonymy) or same words have different meaning in different contexts (polysemy). Syntactic variation: linked to the multiword problem, such word constructions that are structurally different but semantically same, like considering certain language barriers and consideration of these language barriers both could reduce to considering language barriers. Stemming can have negative effect on the performance traceability recovery approaches. Stemming can improve the performance a little bit when size of corpus is small, but it negatively degraded the precision when corpus is of large size [8]. M.F Porters stemmer is the most widely used have drawbacks like handling named entities that would also be stemmed unnecessarily hence leaving data in inconsistent state [62]. Thesaurus creation is manual and not always useful to produce desired results. Some stubborn traces are very difficult to trace even with the help of a thesaurus because language of document does not match with the synonyms defined in the thesaurus [72]. They suggested that use of general purpose thesaurus may increase recall and precision for some datasets, but such improvements are very minor and not consistent across different datasets. Named entities is another deficiency in thesaurus-based traceability recovery approaches. Refactoring consists of a series of small transformations to restore lexical structure of an artifact. Refactoring may be manual, semi-automated or fully automated. In manual refactoring software engineers analyze code for bad smells to suggest changes to

improve quality of code, it is described as error prone and time consuming due to its manual nature [58]. In semi-automated refactoring, software engineer carries out the activities like locating entities while applying transformations

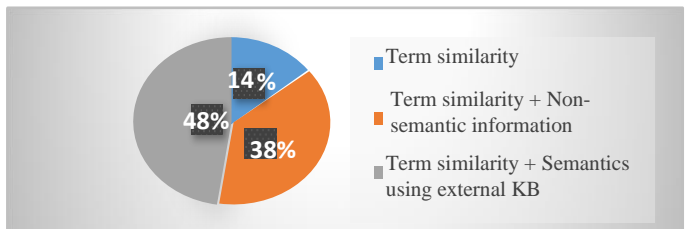


Figure 5: Distribution of Studies

with the help of automated tools. While automated refactoring includes, identification of bad smells from code then applying required transformation automatically, even in this type of refactoring final decision of accepting or rejecting the outcome of transformation is left to human [43]. In [53] authors identified three types of refactoring techniques (restore, remove and move textual information from an artifact) that can support automated traceability.

User feedback approach tends to solve synonymy and polysemy problems. Term weight can be increased or decreased according to the occurrence of that term in a link rejected or confirmed by the user [41]. Negative feedback can be used to remove some terms from query, hence improving the query

quality [60]. Another study also conforms similar results but also highlighted the issue that relevance feedbacks can improve the precision and recall only for few iterations. De Lucia, in their study [22] suggested that if you want to retrieve all links among different software artifacts, relevance feedback will not improve the performance of IR method. They also proved that performance of relevance feedback is dataset dependent and varies over different recall threshold values for same data-set. [13] highlights some of the limitations of relevance feedback, first it requires that query or source artifact should contain few words rather than the target artifact because standard Rocchio tends to increase the size of query by adding terms from relevant documents. Secondly it requires relevant documents to be in

They used the less verbose artifact as query to overcome the first limitation and to avoid second limitation they apply relevance feedback only when number of true positive links are equal to or greater than the number of false positives. But if we want to achieve 100% recall, then for a corpus like easy clinic that have 30 use cases traced to 47 code classes, one have to analyze about 1000 trace links to find 93 correct links, which is a definite limitation not mitigated by the approaches like feedback analysis [22], [48]. LSI is an extension of vector space model [10] which considers the associations between terms and documents. The idea behind LSI is that it assumes that there is a hidden structure in usage of words hidden because of variability in vocabulary of artifacts. Statistical techniques can be used to approximate this latent structure. LSI uses this latent semantic structure for formulating queries and describing documents to overcome the problems like synonymy without the need of a thesaurus, stemming or other such technique [21]. [63] proved that performance of LSI heavily dependent upon the choice of the value of a constant K and on similarity ranking between term and document called threshold value. Use of query expansion techniques proved to be very effective in traceability recovery techniques. Query expansion technique can be used to expand the query artifact with semantically similar terms learned through web for addressing term mismatch problem [40]. Similarly [51] uses semantically similar terms extracted from sources such as Wikipedia to expand the corpus. [16] traced HIPPA regulations with 10 health care systems. They manually search the web (BING, GOOGLE, and YAHOO) by using titles of each regulation. Top ten terms form ten most relevant documents were selected to expand each HIPAA regulation and perform traceability using VSM. [35] extracted stubborn trace links in similar way but instead of processing full documents they identify related chunks of texts from a document to mine domain specific terms. Modifying the trace query using this approach has been shown to improve the recall and precision of certain stubborn trace links. But Wikipedia and other online knowledge sources are not trusted sources as they don't represent any business models, adding terms from them can worsen the performance of trace link [50]. In [17] authors also used relevant documentation to expand the source artifact. They apply VSM with other IR models but in some cases, it degraded the performance rather than improving it. In [50] authors used natural language processing techniques to extract subject, object and predicate terms by use of algorithms called triples extraction algorithm and used DBpedia to extract categories of triplets to expand the query artifact. Machine learning approach can be used to expand the query artifact, but this approach requires training dataset which is not necessarily available [16]. Table IV recaps the advantages and limitations of the approaches and Table V summarize the techniques with respect to type of term mismatch problem it intends to solve.

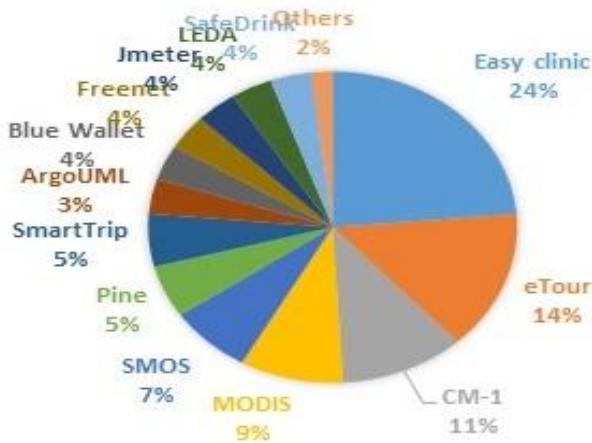


Figure 6: Dataset usage in literature

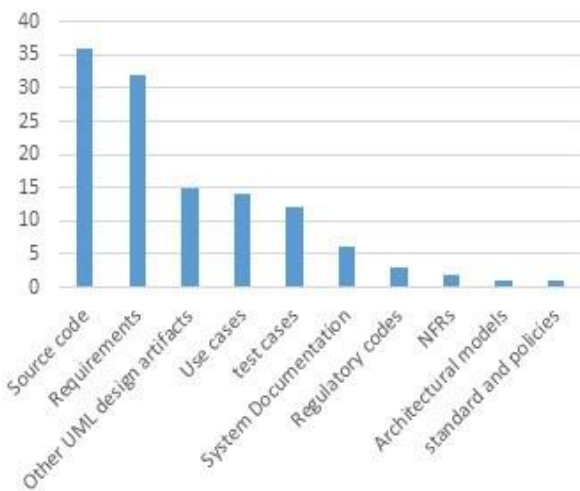


Figure 7: Frequency of software artifacts used in literature to demonstrate traceability approach

same cluster which is not always possible.

An adaptive version of relevance feedback [60] was suggested to overcome the limitations of relevance feedback.

TABLE IV: ADVANTAGES AND LIMITATIONS OF THE APPROACHES

Technique	Advantage	Limitation
Stemming	Tool support, can handle morphological variations only.	Over stemming reduces precision. Under stemming reduces recall.
Thesaurus	To handle semantic variations. Provide solution for synonymy and polysemy problem. Provide a restricted vocabulary to be used in the system.	General purpose thesaurus has Completeness issues they are of very large size. Domain specific thesaurus are difficult to build and Costly. Named entities problem
Refactoring	Improves the code by restoring textual structure therefore making it more feasible for information retrieval.	Manual. A difficult technique that can introduce bugs in the system. Moreover, refactoring is mostly used only for normalizing source code vocabulary.
User feedback	Can solve synonymy, polysemy problems.	User must analyze each link manually. Human dependent technique.
LSI	Use of surrounding words of matching terms can accommodate some of mismatching terms. It is a distinct feature of LSI.	High computational cost. Finding optimal configuration of LSI is NP complete problem. Performance of LSI greatly varied in different corpuses.
Query expansion	Increase recall. Unconditional query expansion reduces precision. Deals with semantic variations like synonyms and polysems.	External knowledge sources are not trusted as. Using world wide web as external source has its own limitation as selecting documents from web is a manual process, otherwise lot of noise could be included in the query.
Natural language processing	Can take advantage of phrases rather than terms and context of the terms can be extracted by NLP.	NLP techniques had been used along with external KB which are not trusted.

TABLE V: CLASSIFICATION OF TECHNIQUES BASED ON TERM MISMATCH PROBLEM IT SOLVE.

Type of Variation	Problem	Solutions
Morphological variations	Root word problem. Abbreviations, acronyms.	Stemming
Semantic variations	Polysemy, synonymy, hyponymy.	Thesaurus, External knowledge sources, query expansion.
Syntactic variation.	Multi word problems.	Use of word ordering and phrasing.

IV. VALIDITY THREATS

Multiple threats are of relevance here, which are shortly discussed:

- *Missing relevant studies:* Our search focused on requirement traceability solutions based on IR techniques

Thus, some studies from the population might have been missed. However, the search returned a high number of relevant studies with little noise. This indicates that the systematic search helped to cover a larger portion of the population. Because of context specific search terms, there is a risk of missing studies from the population.

- *Bias in selection:* Study selection, data extraction and analysis activities are prone to biases and mistakes. To avoid any bias or mistake in selection, two researchers have been involved and cross-checking of the studies has been done among the authors of the paper. The use of a research protocol is also a means of reducing bias. To further avoid the risk, the research protocol was evaluated by an independent researcher.

- *Wrong Conclusions:* To reduce the risk of drawing wrong implications and key findings were discussed among the authors. We believe this helped us in reducing the risk of drawing wrong conclusions from the data.

V. CONCLUSION

We presented a systematic literature review to, 1) classify the IR based traceability recovery techniques, 2) highlight the advantages and limitations of these techniques, 3) classify the dataset used by the techniques, 4) identify the software artifacts used to demonstrate the traceability approaches, and 5) identify the approaches solving the term mismatch issue. The results of our study show that during IR based traceability recovery the term mismatch problem is a major barrier. There are different aspects of term mismatch problem. The studies in this area addressed only few aspects of this problem. Only one study takes care of word ordering during term matching and all other studies used bag of words approach. Some studies solve the synonyms problem with the help of external knowledge base which is slower in executing commands and used too much space. The most widely used approach to solve synonymy and polysemy problem is query expansion (48%). But these studies expand the query unconditionally bringing too much noise into the query artifact. This noise could result in reduced precision. Query expansion technique had the potential to solve the term mismatch problem. Further investigation is needed in this area to find the mismatching terms from query artifact and expand only these terms, precision could be improved in this way. Thesaurus based approaches acquire good precision but cost of building a domain specific thesaurus is very high. Automating the thesaurus creation could overcome this limitation making thesaurus-based approaches useful.

REFERENCES

- [1] Aharon Abadi, Mordechai Nisenson, and Yahalomit Simionovici. A traceability technique for specifications. In *Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on*, pages 103–112. IEEE, 2008.
- [2] Nicolas Anquetil and Timothy Lethbridge. Assessing the relevance of identifier names in a legacy software system. In *Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative Research*, page 4. IBM Press, 1998.
- [3] Giuliano Antoniol, Gerardo Canfora, Gerardo Casazza, Andrea De Lucia, and Ettore Merlo. Recovering traceability links between code and documentation. *IEEE transactions on software engineering*, 28(10):970–983, 2002.
- [4] Giuliano Antoniol, Bruno Caprile, Alessandra Potrich, and Paolo Tonella. Design-code traceability recovery: selecting the basic linkage properties. *Science of Computer Programming*, 40(2-3):213–234, 2001.
- [5] Hazeline U Asuncion, Arthur U Asuncion, and Richard N Taylor. Software traceability with topic modeling. In *Software Engineering, 2010 ACM/IEEE 32nd International Conference on*, volume 1, pages 95–104. IEEE, 2010.
- [6] Alberto Bacchelli, Michele Lanza, and Romain Robbes. Linking e-mails and source code artifacts. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 375–384. ACM, 2010.
- [7] Gabriele Bavota, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, Fabio Ricci, and Genoveffa Tortora. The role of artefact corpus in Isi-based traceability recovery. In *Traceability in Emerging Forms of Software Engineering (TEFSE), 2013 International Workshop on*, pages 83–89. IEEE, 2013.
- [8] Gabriele Bavota, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, Fabio Ricci, and Genoveffa Tortora. The role of artefact corpus in Isi-based traceability recovery. In *Traceability in Emerging Forms of Software Engineering (TEFSE), 2013 International Workshop on*, pages 83–89. IEEE, 2013.
- [9] Gabriele Bavota, Andrea De Lucia, Rocco Oliveto, and Genoveffa Tortora. Enhancing software artefact traceability recovery processes with link count information. *Information and Software Technology*, 56(2):163–182, 2014.
- [10] Giovanni Capobianco, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, and Sebastiano Panichella. On the role of the nouns in ir-based traceability recovery. In *Program Comprehension, 2009. ICPC'09. IEEE 17th International Conference on*, pages 148–157. IEEE, 2009.
- [11] Giovanni Capobianco, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, and Sebastiano Panichella. Traceability recovery using numerical analysis. In *Reverse Engineering, 2009. WCRE'09. 16th Working Conference on*, pages 195–204. IEEE, 2009.
- [12] Giovanni Capobianco, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, and Sebastiano Panichella. Improving ir-based traceability recovery via noun-based indexing of software artifacts. *Journal of Software: Evolution and Process*, 25(7):743–762, 2013.
- [13] Stefano Ceri, Alessandro Bozzon, Marco Brambilla, Emanuele Della Valle, Piero Fraternali, and Silvia Quarteroni. An introduction to information retrieval. In *Web information retrieval*, pages 3–11. Springer, 2013.
- [14] Xiaofan Chen. Extraction and visualization of traceability relationships between documents and source code. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, pages 505–510. ACM, 2010.
- [15] Jane Cleland-Huang, Brian Berenbach, Stephen Clark, Raffaella Settini, and Eli Romanova. Best practices for automated traceability. *Computer*, 40(6), 2007.
- [16] Jane Cleland-Huang, Adam Czauderna, Marek Gibiec, and John Emenecker. A machine learning approach for tracing regulatory codes to product specific requirements. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 155–164. ACM, 2010.
- [17] Tathagata Dasgupta, Mark Grechanik, Evan Moritz, Bogdan Dit, and Denys Poshyvanyk. Enhancing software traceability by automatically expanding corpora with relevant documentation. In *2013 IEEE International Conference on Software Maintenance*, pages 320–329. IEEE, 2013.
- [18] Tathagata Dasgupta, Mark Grechanik, Evan Moritz, Bogdan Dit, and Denys Poshyvanyk. Enhancing software traceability by automatically expanding corpora with relevant documentation. In *2013 IEEE International Conference on Software Maintenance*, pages 320–329. IEEE, 2013.
- [19] Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, Annibale Panichella, and Sebastiano Panichella. Improving ir-based traceability recovery using smoothing filters. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on*, pages 21–30. IEEE, 2011.
- [20] Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, Annibale Panichella, and Sebastiano Panichella. Applying a smoothing filter to improve ir-based traceability recovery processes: An empirical investigation. *Information and Software Technology*, 55(4):741–754, 2013.
- [21] Andrea De Lucia, Fausto Fasano, Rocco Oliveto, and Genoveffa Tortora. Enhancing an artefact management system with traceability recovery features. In *Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on*, pages 306–315. IEEE, 2004.
- [22] Andrea De Lucia, Rocco Oliveto, and Paola Sgueglia. Incremental approach and user feedbacks: a silver bullet for traceability recovery. In *Software Maintenance, 2006. ICSM'06. 22nd IEEE International Conference on*, pages 299–309. IEEE, 2006.
- [23] Andrea De Lucia, Rocco Oliveto, and Genoveffa Tortora. Assessing ir-based traceability recovery tools through controlled experiments. *Empirical Software Engineering*, 14(1):57–92, 2009.
- [24] Fangshu Di and Maolin Zhang. An improving approach for recovering requirements-to-design traceability links. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1–6. IEEE, 2009.
- [25] Andre Di Thommazo, Thiago Ribeiro, Guilherme Olivatto, Vera Werneck, and Sandra Fabbri. An automatic approach to detect traceability links using fuzzy logic. In *Software Engineering (SBES), 2013 27th Brazilian Symposium on*, pages 21–30. IEEE, 2013.
- [26] Diana Diaz, Gabriele Bavota, Andrian Marcus, Rocco Oliveto, Silvia Takahashi, and Andrea De Lucia. Using code ownership to improve ir-based traceability link recovery. In *Program Comprehension (ICPC), 2013 IEEE 21st International Conference on*, pages 123–132. IEEE, 2013.
- [27] Russell C Eberhart, Yuhui Shi, and James Kennedy. *Swarm intelligence*. Elsevier, 2001.
- [28] Farina Faiz, Rubaida Easmin, and Alim Ul Gias. Achieving better requirements to code traceability: Which refactoring should be done first? In *Quality of Information and Communications Technology (QUATIC), 2016 10th International Conference on the*, pages 9–14. IEEE, 2016.
- [29] Martin Feilkas, Daniel Ratiu, and Elmar Jurgens. The loss of architectural knowledge during system evolution: An industrial case study. In *2009 IEEE 17th International Conference on Program Comprehension (ICPC 2009)*, pages 188–197. IEEE, 2009.
- [30] Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 1999.
- [31] GW Furnas, TK Landauer, LM Gomez, and ST Dumais. The vocabulary problem in human-computer interaction. *Communications of the ACM*, 30(11):964–971, 1987.
- [32] Carmen Galvez, Felix´ de Moya-Anegon,´ and V´ictor H Solana. Term conflation methods in information retrieval: Nonlinguistic and linguistic approaches. *Journal of Documentation*, 61(4):520–547, 2005.
- [33] Gregory Gay, Sonia Haiduc, Andrian Marcus, and Tim Menzies. On the use of relevance feedback in ir-based concept location. In *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*, pages 351–360. IEEE, 2009.
- [34] Achraf Ghabi and Alexander Egyed. Code patterns for automatically validating requirements-to-code traces. In *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on*, pages 200–209. IEEE, 2012.
- [35] Marek Gibiec, Adam Czauderna, and Jane Cleland-Huang. Towards mining replacement queries for hard-to-retrieve traces. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, pages 245–254. ACM, 2010.
- [36] Marek Gibiec, Adam Czauderna, and Jane Cleland-Huang. Towards mining replacement queries for hard-to-retrieve traces. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, pages 245–254. ACM, 2010.

- [37] Orlena Gotel and Anthony Finkelstein. Extended requirements traceability: Results of an industrial case study. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 169–178. IEEE, 1997.
- [38] Jin Guo, Jinghui Cheng, and Jane Cleland-Huang. Semantically enhanced software traceability using deep learning techniques. In *Software Engineering (ICSE), 2017 IEEE/ACM 39th International Conference on*, pages 3–14. IEEE, 2017.
- [39] Jin Guo, Jane Cleland-Huang, and Brian Berenbach. Foundations for an expert system in domain-specific traceability. In *Requirements Engineering Conference (RE), 2013 21st IEEE International*, pages 42–51. IEEE, 2013.
- [40] Jin Guo, Marek Gibiec, and Jane Cleland-Huang. Tackling the term-mismatch problem in automated trace retrieval. *Empirical Software Engineering*, 22(3):1103–1142, 2017.
- [41] Jane Huffman Hayes, Alex Dekhtyar, and Senthil Karthikeyan Sundaram. Advancing candidate link generation for requirements tracing: The study of methods. *IEEE Transactions on Software Engineering*, 32(1):4, 2006.
- [42] Ritika Jain, Smita Ghaisas, and Ashish Sureka. Sanayojan: a framework for traceability link recovery between use-cases in software requirement specification and regulatory documents. In *Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, pages 12–18. ACM, 2014.
- [43] Marija Katic and Kre' simir Fertalj. Towards an appropriate software refactoring tool support. In *Proceedings of the 9th WSEAS international conference on Applied computer science*, pages 140–145. World Scientific and Engineering Academy and Society (WSEAS), 2009.
- [44] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.
- [45] Barbara Kitchenham, Rialethe Pretorius, David Budgen, O Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. Systematic literature reviews in software engineering—a tertiary study. *Information and Software Technology*, 52(8):792–805, 2010.
- [46] Wei-Keat Kong and Jane Huffman Hayes. Proximity-based traceability: An empirical validation using ranked retrieval and set-based measures. In *Empirical Requirements Engineering (EmpiRE), 2011 First International Workshop on*, pages 45–52. IEEE, 2011.
- [47] Adrian Kuhn, Stephane Ducasse, and Tudor G' irba. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230–243, 2007.
- [48] Andrea De Lucia, Fausto Fasano, Rocco Oliveto, and Genoveffa Tortora. Recovering traceability links in software artifact management systems using information retrieval methods. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 16(4):13, 2007.
- [49] Khalid Mahmood, Mazen Alobaidi, and Hironao Takahashi. Autonomous decentralized semantic based traceability link recovery framework. *IEICE TRANSACTIONS on Information and Systems*, 99(9):2283–2294, 2016.
- [50] Khalid Mahmood, Hironao Takahashi, and Mazen Alobaidi. A semantic approach for traceability link recovery in aerospace requirements management system. In *Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on*, pages 217–222. IEEE, 2015.
- [51] Anas Mahmoud. An information theoretic approach for extracting and tracing non-functional requirements. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, pages 36–45. IEEE, 2015.
- [52] Anas Mahmoud and Nan Niu. Supporting requirements traceability through refactoring. In *Requirements Engineering Conference (RE), 2013 21st IEEE International*, pages 32–41. IEEE, 2013.
- [53] Anas Mahmoud and Nan Niu. Supporting requirements to code traceability through refactoring. *Requirements Engineering*, 19(3):309–329, 2014.
- [54] Anas Mahmoud and Nan Niu. On the role of semantics in automated requirements tracing. *Requirements Engineering*, 20(3):281–300, 2015.
- [55] Anas Mahmoud, Nan Niu, and Songhua Xu. A semantic relatedness approach for traceability link recovery. In *Program Comprehension (ICPC), 2012 IEEE 20th International Conference on*, pages 183–192. IEEE, 2012.
- [56] Anas Mahmoud and Grant Williams. Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, 21(3):357–381, 2016.
- [57] Collin McMillan, Denys Poshyvanyk, and Meghan Revelle. Combining textual and structural analysis of software artifacts for traceability link recovery. In *Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering*, pages 41–48. IEEE Computer Society, 2009.
- [58] Erica Mealy, David Carrington, Paul Strooper, and Peta Wyeth. Improving usability of software refactoring tools. In *Software Engineering Conference, 2007. ASWEC 2007. 18th Australian*, pages 307–318. IEEE, 2007.
- [59] Annibale Panichella, Andrea De Lucia, and Andy Zaidman. Adaptive user feedback for ir-based traceability recovery. In *Proceedings of the 8th International Symposium on Software and Systems Traceability*, pages 15–21. IEEE Press, 2015.
- [60] Annibale Panichella, Andrea De Lucia, and Andy Zaidman. Adaptive user feedback for ir-based traceability recovery. In *Proceedings of the 8th International Symposium on Software and Systems Traceability*, pages 15–21. IEEE Press, 2015.
- [61] Annibale Panichella, Collin McMillan, Evan Moritz, Davide Palmieri, Rocco Oliveto, Denys Poshyvanyk, and Andrea De Lucia. When and how using structural information to improve ir-based traceability recovery. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 199–208. IEEE, 2013.
- [62] C Ramasubramanian and R Ramya. Effective pre-processing activities in text mining using improved porters stemming algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12):4536–4538, 2013.
- [63] Roslan Sadjirin, Noli Maishara Nordin, Mohd Ikhsan Md Raus, and Zulazeze Sahri. Investigating the optimise k-dimensions and threshold values of latent semantic indexing retrieval performance for small malay language corpus. In *Regional Conference on Science, Technology and Social Sciences (RCSTSS 2014)*, pages 325–336. Springer, 2016.
- [64] George Spanoudakis, Andrea Zisman, Elena Perez-Minana, and Paul Krause. Rule-based generation of requirements traceability relations. *Journal of Systems and Software*, 72(2):105–127, 2004.
- [65] Darijus Strasunskas and Sari E Hakkarainen. Domain model-driven software engineering: A method for discovery of dependency links. *Information and Software Technology*, 54(11):1239–1249, 2012.
- [66] David R Tobergte and Shirley Curtis. No title no title. *Journal of Chemical Information and Modeling*, 53(9):1689–1699, 2013.
- [67] Ryosuke Tsuchiya, Hironori Washizaki, Yoshiaki Fukazawa, Keishi Oshima, and Ryota Mibe. Interactive recovery of requirements traceability links using user feedback and configuration management logs. In *International Conference on Advanced Information Systems Engineering*, pages 247–262. Springer, 2015.
- [68] Yoshihisa Udagawa. Enhancing information retrieval to automatically trace requirements and design artifacts. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, pages 292–295. ACM, 2011.
- [69] Soha M Yassin, Ibrahim F Moawad, Rania E Gohary, and Mohamed F Tolba. Ontology-based model for rational unified process artifacts traceability. In *Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on*, pages 241–248. IEEE, 2012.
- [70] Yonggang Zhang, Rene Witte, Juergen Rilling, and Volker Haarslev. Ontological approach for the semantic recovery of traceability links between software artefacts. *IET software*, 2(3):185–203, 2008.
- [71] Yuchen Zhang, Chengcheng Wan, and Bo Jin. An empirical study on recovering requirement-to-code links. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on*, pages 121–126. IEEE, 2016.
- [72] Xuchang Zou, Raffaella Settini, and Jane Cleland-Huang. Improving automated requirements trace retrieval: a study of term-based enhancement methods. *Empirical Software Engineering*, 15(2):119–146, 2010.