# Technical Report: A Systematic Mapping Study on the Combination of Information Retrieval and Requirements Traceability: Models, Strategies, Datasets, Measures and Baselines

## 1. Extracted Data

### *1.1.* Extracted Data for RQ1

| Index | Title | Author | Year | Venue | Publication type |
|---|---|---|---|---|---|
| S1 | An empirical study on recovering requirement-to-code links | Zhang Yuchen<br>Wan Chengcheng<br>Jin Bo | 2016 | SNPD | Conference |
| S2 | An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis | Jiale Zhou<br>Yue Lu<br>Kristina Lundqvist | 2013 | Semantic Scholar | Other |
| S3 | Configuring Latent Semantic Indexing for Requirements Tracing | Sebastian Eder<br>Henning Femmer<br>Benedikt Hauptmann<br>Maximilian Junker | 2015 | RET | Conference |
| S4 | A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems | Jiale Zhou<br>Yue Lu<br>Kristina Lundqvist | 2013 | SEAA | Conference |
| S5 | Analyzing closeness of code dependencies for improving IR-based Traceability Recovery | Hongyu Kuang<br>Jia Nie<br>Hao Hu<br>Patrick Rempel<br>Jian Lü<br>Alexander Egyed<br>Patrick Mäder | 2017 | SANER | Conference |
| S6 | An empirical study on the importance of source code entities for requirements traceability | Nasir Ali<br>Zohreh Sharafi<br>Yann-Gaël Guéhéneuc<br>Giuliano Antoniol | 2014 | ESE | Journal |
| S7 | Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery | Haijuan Wang<br>Guohua Shen<br>Zhiqiu Huang<br>Yaoshen Yu<br>Kai Chen | 2021 | FITEE | Journal |
| S8 | A semantic relatedness approach for traceability link recovery | Anas Mahmoud<br>Nan Niu<br>Songhua Xu | 2012 | ICPC | Conference<br>Workshop Papers |
| S9 | Achieving better requirements to code traceability: which refactoring should be done first? | Farina Faiz<br>Rubaida Easmin<br>Alim Ul Gias | 2016 | QUATIC | Conference |
| S10 | An Empirical Study on Requirements Traceability Using Eye-Tracking | Nasir Ali<br>Zohreh Sharafl<br>Yann-Gaël Guéhéneuc<br>Giuliano Antoniol | 2012 | ICSM | Conference |
| S11 | An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing | Jianwei Shao<br>Wei Wu<br>Peng Geng | 2013 | ICCSA | Conference |
| S12 | An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery | Danissa V. Rodriguez<br>Doris L. Carver | 2020 | ICTAI | Conference |
| S13 | Toward multilevel textual requirements traceability using model-driven engineering and information retrieval | Nicolas Sannier<br>Benoit Baudry | 2012 | MoDRE | Conference |
| S14 | Code Patterns for Automatically Validating Requirements-to-Code Traces | Achraf Ghabi<br>Alexander Egyed | 2012 | ASE | Conference |
| S15 | Combining VSM and BTM to improve requirements trace links generation | Bangchao Wang<br>Rong Peng<br>Zhuo Wang | 2019 | SEKE | Conference |

| | | | | | |
|---|---|---|---|---|---|
| | | Yaxin Zhao | | | |
| S16 | Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links | Nasir Ali<br>Yann-Gaël Guéhéneuc<br>Giuliano Antoniol | 2013 | TSE | Journal |
| S17 | Leveraging historical co-change information for requirements traceability | Nasir Ali<br>Fehmi Jaafar<br>Ahmed E. Hassan | 2013 | WCRE | Conference |
| S18 | Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery | Danissa V. Rodriguez<br>Doris L. Carver | 2020 | EIT | Conference |
| S19 | Filtering of false positives from IR-based traceability links among software artifacts | Jyoti<br>Jitender Kumar Chhabra | 2017 | I2CT | Conference |
| S20 | Quality improvements for trace links between source code and requirements | Paul Hübner | 2016 | REFSQ | Conference |
| S21 | Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts | Shouichi Nagano<br>Yusuke Ichikawa<br>Toru Kobayashi | 2012 | COMPSAC | Conference |
| S22 | Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling | Jyoti<br>Jitender Kumar Chhabra | 2017 | ICAICR | Journal |
| S23 | An Automated Hybrid Approach for Generating Requirements Trace Links | Wang Bangchao<br>Peng Rong<br>Wang Zhuo<br>Wang Xiaomin<br>Li Yuanbang | 2020 | IJSEKE | Journal |
| S24 | Enhancing candidate link generation for requirements tracing：The cluster hypothesis revisited | Nan Niu<br>Anas Mahmoud | 2012 | RE Conference | Conference and Workshop Papers |
| S25 | Search-Based Requirements Traceability Recovery： A Multi-Objective Approach | Adnane Ghannem<br>Mohamed Salah Hamdi<br>Marouane Kessentini<br>Hany H. Ammar | 2017 | CEC | Conference and Workshop Papers |
| S26 | Supporting requirements traceability through refactoring | Anas Mahmoud<br>Nan Niu | 2013 | RE Conference | Journal Articles |
| S27 | SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents | Ritika Jain<br>Smita Ghaisas<br>Ashish Sureka | 2014 | RAISE | Conference |
| S28 | Towards feature-aware retrieval of refinement traces | Patrick Rempel<br>Patrick Mäder<br>Tobias Kuschke | 2013 | TEFSE | Conference |
| S29 | Interactive recovery of requirements traceability links using user feedback and configuration management logs | Ryosuke Tsuchiya<br>Hironor Washizakii<br>Yoshiaki Fukazawa<br>Keishi Oshima<br>Ryota Mibe | 2015 | CAiSE | Conference |
| S30 | Supporting requirements to code traceability through refactoring | Anas Mahmoud<br>Nan Niu | 2014 | RE Conference | Journal |
| S31 | Recovering traceability links between requirements and source code using the configuration management log | Ryosuke Tsuchiya<br>Hironori Washizaki<br>Yoshiaki Fukazawa<br>Tadahisa Kato<br>Masumi Kawakami<br>Kentaro Yoshimura | 2015 | IEICE | Journal |
| S32 | Exploiting Parts-of-Speech for Effective Automated Requirements Traceability | Nasir Ali<br>Haipeng Cai<br>Abdelwahab Hamou-Lhadj<br>Jameleddine Hassine | 2018 | IST | Journal |
| S33 | Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery | Hongyu Kuang<br>Hui Gao<br>Hao Hu<br>Xiaoxing Ma | 2019 | ICPC | Conference |

| | | | | | |
|---|---|---|---|---|---|
| | | Jian Lü<br>Patrick Mäder<br>Alexander Egyed | | | |
| S34 | Adaptive User Feedback for IR-Based Traceability Recovery | Annibale Panichella<br>Andrea De Lucia<br>Andy Zaidman | 2015 | SST | Conference |
| S35 | Evolving Software Trace Links between Requirements and Source Code | Mona Rahimi<br>Jane Cleland-Huang | 2018 | ESE | Journal |
| S36 | Supporting Requirements to Code Traceability Creation by Code Comments | Guohua Shen<br>Haijuan Wang<br>Zhiqiu Huang<br>YaoShen Yu<br>Kai Chen | 2021 | IJSEKE | Journal |
| S37 | Using code ownership to improve IR-based Traceability Link Recovery | Diana Diaz<br>Gabriele Bavota<br>Andrian Marcus<br>Rocco Oliveto<br>Silvia Takahashi<br>Andrea De Lucia | 2013 | ICPC | Conference |

*1.2.* Extracted Data for RQ2

| Index | Title | IR Model | Stage |
|---|---|---|---|
| S1 | An empirical study on recovering requirement-to-code links | VSM | Links Generation Stage |
| S2 | An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis | VSM | Preprocessing Stage |
| S3 | Configuring Latent Semantic Indexing for Requirements Tracing | LSI | Links Generation Stage |
| S4 | A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems | VSM | Preprocessing Stage |
| S5 | Analyzing closeness of code dependencies for improving IR-based Traceability Recovery | VSM<br>LSI<br>JS | Links Refinement Stage |
| S6 | An empirical study on the importance of source code entities for requirements traceability | LSI<br>TM(LDA) | Preprocessing Stage |
| S7 | Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery | VSM | Links Generation Stage |
| S8 | A semantic relatedness approach for traceability link recovery | VSM<br>LSI | Preprocessing Stage |
| S9 | Achieving better requirements to code traceability：which refactoring should be done first? | VSM<br>LSI | Preprocessing Stage |
| S10 | An Empirical Study on Requirements Traceability Using Eye-Tracking | LSI<br>TM(LDA) | Preprocessing Stage |
| S11 | An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing | LSI | Preprocessing Stage<br>Links Generation Stage |
| S12 | An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery | IR-based+ | Links Generation Stage |
| S13 | Toward multilevel textual requirements traceability using model-driven engineering and information retrieval | VSM | Links Generation Stage |
| S14 | Code Patterns for Automatically Validating Requirements-to-Code Traces | TM(RTM) | Links Generation Stage |
| S15 | Combining VSM and BTM to improve requirements trace links generation | VSM<br>TM(BTM) | Links Generation Stage |
| S16 | Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links | VSM<br>JS | Links Generation Stage |
| S17 | Leveraging historical co-change information for requirements traceability | VSM<br>JS | Links Generation Stage |
| S18 | Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery | IR-based+ | Links Generation Stage |
| S19 | Filtering of false positives from IR-based traceability links among software artifacts | VSM | Links Refinement Stage |

| | | | |
|---|---|---|---|
| S20 | Quality improvements for trace links between source code and requirements | IR-based+ | Links Generation Stage |
| S21 | Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts | LSI | Preprocessing Stage |
| S22 | Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling | VSM | Links Refinement Stage |
| S23 | An Automated Hybrid Approach for Generating Requirements Trace Links | VSM<br>TM(BTM) | Links Generation Stage |
| S24 | Enhancing candidate link generation for requirements tracing：The cluster hypothesis revisited | VSM | Links Refinement Stage |
| S25 | Search-Based Requirements Traceability Recovery： A Multi-Objective Approach | IR-based+ | Links Generation Stage |
| S26 | Supporting requirements traceability through refactoring | VSM | Preprocessing Stage |
| S27 | SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents | TM(LDA) | Preprocessing Stage,<br>Links Generation Stage |
| S28 | Towards feature-aware retrieval of refinement traces | VSM<br>LSI | Links Refinement Stage |
| S29 | Interactive recovery of requirements traceability links using user feedback and configuration management logs | VSM | Links Generation Stage Preprocessing Stage |
| S30 | Supporting requirements to code traceability through refactoring | VSM<br>LSI | Preprocessing Stage |
| S31 | Recovering traceability links between requirements and source code using the configuration management log | VSM | Links Generation Stage<br>Preprocessing Stage |
| S32 | Exploiting Parts-of-Speech for Effective Automated Requirements Traceability | VSM<br>JS | Links Refinement Stage |
| S33 | Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery | VSM<br>LSI<br>JS | Links Refinement Stage |
| S34 | Adaptive User Feedback for IR-Based Traceability Recovery | VSM | Links Refinement Stage |
| S35 | Evolving Software Trace Links between Requirements and Source Code | VSM<br>LSI | Links Generation Stage |
| S36 | Supporting Requirements to Code Traceability Creation by Code Comments | VSM | Preprocessing Stage |
| S37 | Using code ownership to improve IR-based Traceability Link Recovery | VSM<br>BM | Links Generation Stage |

### *1.3.* Extracted Data for RQ3

| Index | Title | Enhancement strategy |
|---|---|---|
| S1 | An empirical study on recovering requirement-to-code links | Verb-object Phrases |
| S2 | An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis | Context-based |
| S3 | Configuring Latent Semantic Indexing for Requirements Tracing | None |
| S4 | A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems | Context-based |
| S5 | Analyzing closeness of code dependencies for improving IR-based Traceability Recovery | Analyzing Close Relations of Code Dependencies |
| S6 | An empirical study on the importance of source code entities for requirements traceability | Improved Term Weighting Scheme |
| S7 | Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery | Analyzing Close Relations |
| S8 | A semantic relatedness approach for traceability link recovery | None |
| S9 | Achieving better requirements to code traceability：which refactoring should be done first? | Refactoring |
| S10 | An Empirical Study on Requirements Traceability Using Eye-Tracking | Term weighting scheme |
| S11 | An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing | Term Classification<br>Class Clustering |
| S12 | An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery | None |
| S13 | Toward multilevel textual requirements traceability using model-driven engineering and information retrieval | Model-Driven Engineering (MDE) |
| S14 | Code Patterns for Automatically Validating Requirements-to-Code Traces | Code Calling Relationships |
| S15 | Combining VSM and BTM to improve requirements trace links generation | Hybrid Method |
| S16 | Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links | Mining Software Repositories |

| | | |
|---|---|---|
| S17 | Leveraging historical co-change information for requirements traceability | Historical Co-change Information |
| S18 | Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery | None |
| S19 | Filtering of false positives from IR-based traceability links among software artifacts | Correlation Among Classes |
| S20 | Quality improvements for trace links between source code and requirements | None |
| S21 | Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts | Syntax Tree |
| S22 | Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling | Dynamic Integration of Structural Co-change Coupling |
| S23 | An Automated Hybrid Approach for Generating Requirements Trace Links | Hybrid Method<br>Genetic Algorithm |
| S24 | Enhancing candidate link generation for requirements tracing：The cluster hypothesis revisited | Cluster hypothesis |
| S25 | Search-Based Requirements Traceability Recovery： A Multi-Objective Approach | None |
| S26 | Supporting requirements traceability through refactoring | Refactoring |
| S27 | SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents | None |
| S28 | Towards feature-aware retrieval of refinement traces | Graph Clustering |
| S29 | Interactive recovery of requirements traceability links using user feedback and configuration management logs | Configuration Management Log<br>User Feedback |
| S30 | Supporting requirements to code traceability through refactoring | Refactoring |
| S31 | Recovering traceability links between requirements and source code using the configuration management log | Configuration Management Log<br>Commonality and Variability Analysis（CVA）<br>Classification |
| S32 | Exploiting Parts-of-Speech for Effective Automated Requirements Traceability | ConPOS approach |
| S33 | Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery | Frugal User Feedback with Closeness Analysis on Code |
| S34 | Adaptive User Feedback for IR-Based Traceability Recovery | Adaptive User Feedback |
| S35 | Evolving Software Trace Links between Requirements and Source Code | Trace Link Evolver |
| S36 | Supporting Requirements to Code Traceability Creation by Code Comments | Code Comments |
| S37 | Using code ownership to improve IR-based Traceability Link Recovery | code OwNership |

*1.4.* Extracted Data for RQ4

| **Index** | **Title** | Source Artifact | Target Artifact | **Datasets** |
|---|---|---|---|---|
| S1 | An empirical study on recovering requirement-to-code links | Requirements | Code | eTour<br>iBooks<br>SMS<br>EasyClinic |
| S2 | An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis | Use Cases | Code | eTour<br>iTrust |
| S3 | Configuring Latent Semantic Indexing for Requirements Tracing | Requirements<br>Defect Reports<br>Use Cases<br>Change Requests | Requirements<br>Use Cases<br>Test Cases<br>Use Cases | MODIS<br>CM-1<br>EasyClinic<br>MR0<br>MR1<br>MR2 |
| S4 | A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems | Use Cases | Code | iRobot<br>iTruck<br>iSudoku |
| S5 | Analyzing closeness of code dependencies for improving IR-based Traceability Recovery | Requirements | Code | iTrust<br>GanttProject<br>jHotDraw |
| S6 | An empirical study on the importance of source code entities for requirements traceability | Requirements | Code | iTrust<br>Lucene<br>Pooka |
| S7 | Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery | Use Case<br>Requirements | Test Case<br>Design | EasyClinic<br>CM1-NASA |

| | | | | |
|---|---|---|---|---|
| | | Requirements<br>Requirements<br>Use Case | Use Case<br>Requirements<br>Code | Pine<br>GANNT<br>iTrust |
| S8 | A semantic relatedness approach for traceability link recovery | Use Cases<br>Requirements | Code<br>Design | iTrust<br>eTour<br>CM-1 |
| S9 | Achieving better requirements to code traceability: which refactoring should be done first? | Use Cases | Code | iTrust<br>eTour |
| S10 | An Empirical Study on Requirements Traceability Using Eye-Tracking | Requirements | Code | iTrust<br>Pooka |
| S11 | An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing | Requirements | Code | Labor Market Monitoring Software Product Line (LMMSPL) engineering |
| S12 | An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery | Requirements<br>Use Cases | Code | EBT<br>Albergate<br>eTour |
| S13 | Toward multilevel textual requirements traceability using model-driven engineering and information retrieval | Requirements(not detail) | Requirements(not detail) | None |
| S14 | Code Patterns for Automatically Validating Requirements-to-Code Traces | Requirements | Code | VideoOnDemand (VoD)<br>Chess<br>GanttProject<br>jHotDraw (JHD) |
| S15 | Combining VSM and BTM to improve requirements trace links generation | Use case<br>Requirements<br>Requirements | Test Case<br>Test Case<br>Requirements | WARC<br>EasyClinic<br>EBT |
| S16 | Trustrac: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links | Requirements | Code | jEdit<br>Pooka<br>Rhino<br>SIP Communicator |
| S17 | Leveraging historical co-change information for requirements traceability | Requirements | Code | iTrust<br>Pooka<br>SIP Communicator |
| S18 | Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery | Requirements | Code | EBT<br>Albergate<br>eTour |
| S19 | Filtering of false positives from IR-based traceability links among software artifacts | Use Cases | Code | iTrust |
| S20 | Quality improvements for trace links between source code and requirements | Requirements<br>Use Cases | Code | Mylyn<br>iTrust |
| S21 | Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts | Function Specification | Code | The experiment used actual enterprise project, the project was the NTT search system "i-mage" |
| S22 | Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling | Use Cases | Code | iTrust |
| S23 | An Automated Hybrid Approach for Generating Requirements Trace Links | Use Case<br>Requirements<br>Requirements<br>Use Cases | Test Case<br>Test Case<br>Requirements<br>Code | WARC subset 1<br>WARC subset 2<br>EBT subset<br>EasyClinic<br>eTour |
| S24 | Enhancing candidate link generation for requirements tracing: The cluster hypothesis revisited | Requirements<br>Requirements | Code<br>Design | iTrust<br>eTour<br>CM-1 |
| S25 | Search-Based Requirements Traceability Recovery: A Multi-Objective Approach | Requirements | Code | LEDA<br>Albergate<br>ETOUR |

| | | | | |
|---|---|---|---|---|
| S26 | Supporting requirements traceability through refactoring | Requirements | Code | iTrust<br>eTour<br>WDS |
| S27 | SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents | Use Cases | Regulatory Documents | The experiments on real-world data obtained from software projects of a large global Information Technology (IT) services company |
| S28 | Towards feature-aware retrieval of refinement traces | Requirements<br>Use Case | Use Case<br>Test Case | EasyClinic<br>CM-1<br>Waterloo |
| S29 | Interactive recovery of requirements traceability links using user feedback and configuration management logs | Requirements | Code | Author carried out experiments targeting an enterprise system developed by a Japanese company |
| S30 | Supporting requirements to code traceability through refactoring | Requirements | Code | iTrust<br>eTour<br>WDS |
| S31 | Recovering traceability links between requirements and source code using the configuration management log | Requirements | Code | CUnit<br>Network Control System |
| S32 | Exploiting Parts-of-Speech for Effective Automated Requirements Traceability | Requirements | Code | iTrust<br>Lynx<br>Pooka<br>SIP Communicator |
| S33 | Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery | Requirements | Code | iTrust<br>Maven<br>Pig<br>GanttProject<br>Infinispan |
| S34 | Adaptive User Feedback for IR-Based Traceability Recovery | Use Cases<br>Test Cases<br>Code<br>UML Interaction Diagram<br>Requirements | Code<br>Code<br>JSP<br>Code<br>Requirements | Easy-Clinic<br>i-Trust<br>Modis |
| S35 | Evolving Software Trace Links between Requirements and Source Code | Requirements | Code | Domain Analysis App<br>DOTS File Generator<br>Apache Cassandra Database System<br>Dronology system |
| S36 | Supporting Requirements to Code Traceability Creation by Code Comments | Use Cases | Code | eTour<br>iTrust |
| S37 | Using code ownership to improve IR-based Traceability Link Recovery | Use Cases | Code | eTour<br>SMOS |

*1.5.* Extracted Data for RQ5, RQ6, RQ7 and RQ8

| Index | Title | Intercept points | Measure | Baseline | Evidence level |
|---|---|---|---|---|---|
| S1 | An empirical study on recovering requirement-to-code links | Threshold | Recall<br>Precision<br>F-Measure | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S2 | An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis | Unclear | MAP<br>Precision | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S3 | Configuring Latent Semantic Indexing for Requirements Tracing | Unclear | MAP<br>AP | None | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S4 | A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems | Unclear | Recall,<br>Precision,<br>MAP | VSM | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S5 | Analyzing closeness of code dependencies for improving IR-based Traceability Recovery | Thresholds | Recall<br>Precision<br>F-Measure,<br>MAP<br>AP | VSM<br>LIS<br>JS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S6 | An empirical study on the importance of source | Threshold | Recall | LSI | Level 4. Evidence obtained from |

| | | | | | |
|---|---|---|---|---|---|
| | code entities for requirements traceability | | Precision<br>F-Measure | | industrial studies (e.g., causal case studies in an industrial setting). |
| S7 | Analyzing close relations between target artifacts for<br>improving IR-based requirement traceability recovery | Threshold | Recall<br>Precision<br>MAP<br>AP | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S8 | A semantic relatedness approach for traceability link recovery | Threshold | Recall<br>Precision<br>F-Measure,<br>MAP<br>DiffAR<br>Lag | VSM<br>LSI | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S9 | Achieving better requirements to code traceability：which refactoring should be done first ？ | Threshold | None | VSM<br>LIS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S10 | An Empirical Study on Requirements Traceability Using Eye-Tracking | Threshold | Recall,<br>Precision<br>F-Measure | LSI | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S11 | An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing | Threshold | Recall<br>Precision | LSI | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S12 | An IR-Based Artificial Bee Colony Approach for Traceability Link Recovery | Iteration | Recall<br>Precision<br>F-Measure | IR-based+ | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S13 | Toward multilevel textual requirements traceability using model-driven engineering and information retrieval | None | None | None | Level 1. Evidence obtained from demonstration or working out with toy examples. |
| S14 | Code Patterns for Automatically Validating Requirements-to-Code Traces | None | Recall<br>Precision | TM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S15 | Combining VSM and BTM to improve requirements trace links generation | Threshold | Recall<br>Precision | VSM<br>TM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S16 | Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links | Threshold | Recall<br>Precision<br>F-Measure | VSM<br>JS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S17 | Leveraging historical co-change information for requirements traceability | Threshold | Recall<br>Precision | VSM<br>JS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S18 | Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery | Iteration | Recall<br>Precision<br>F-Measure | VSM<br>LSI<br>JS<br>TM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S19 | Filtering of false positives from IR-based traceability links among software artifacts | Iteration | Recall<br>Precision<br>F-Measure | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S20 | Quality improvements for trace links between source code and requirements | Threshold | Recall<br>Precision | IR-based+ | Level 1. Evidence obtained from demonstration or working out with toy examples. |
| S21 | Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts | Threshold | Recall<br>Precision<br>F-Measure | LIS | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S22 | Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling | Threshold | Recall,<br>Precision<br>F-Measure | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S23 | An Automated Hybrid Approach for Generating Requirements Trace Links | Selectivity | Recall<br>Precision<br>F-Measure<br>Selectivity | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S24 | Enhancing candidate link generation for requirements tracing：The cluster hypothesis revisited | Threshold | Recall<br>Precision<br>MAP | VSM | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S25 | Search-Based Requirements Traceability Recovery： A Multi-Objective Approach | Iteration | Recall<br>Precision | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S26 | Supporting requirements traceability through refactoring | Threshold | Recall<br>Precision MAP<br>DiffAR | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S27 | SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents | Unclear | AP<br>MAP | LDA | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S28 | Towards feature-aware retrieval of refinement traces | Unclear | Recall<br>Precision<br>AP | VSM<br>LSI | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |

| | | | | | |
|---|---|---|---|---|---|
| S29 | Interactive recovery of requirements traceability links using user feedback and configuration management logs | Threshold | Recall<br>Precision<br>F-Measure | VSM | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S30 | Supporting requirements to code traceability through refactoring | Threshold | Recall<br>Precision<br>MAP<br>DiffAR | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S31 | Recovering traceability links between requirements and source code using the configuration management log | Threshold | Recall<br>Precision<br>F-Measure | VSM | Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting). |
| S32 | Exploiting Parts-of-Speech for Effective Automated Requirements Traceability | Unclear | Recall<br>Precision<br>MAP<br>AP | VSM<br>JS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S33 | Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery | Threshold | Recall<br>Precision<br>F-Measure<br>MAP<br>AP | VSM<br>LSI<br>JS | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S34 | Adaptive User Feedback for IR-Based Traceability Recovery | Threshold | Recall<br>Precision | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S35 | Evolving Software Trace Links between Requirements and Source Code | Unclear | Recall<br>Precision<br>F-Measure | VSM<br>LSI | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S36 | Supporting Requirements to Code Traceability Creation by Code Comments | Unclear | Recall<br>Precision MAP<br>AP | VSM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |
| S37 | Using code ownership to improve IR-based Traceability Link Recovery | Threshold | Recall<br>Precision | VSM<br>BM | Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments). |

**Table 7.** List of enhancement strategies for IR-based requirements trace recovery approaches

| Strategy | IR model | | | | | | | Applying Phrase | Strategy Characteristics |
|---|---|---|---|---|---|---|---|---|---|
| | VSM | LSI | JS | LDA | BTM | RTM | BM | | |
| Context- based [S2][S4] | ● | | | | | | | P | Separating intent from context in requirements |
| Improved Term Weighting Scheme [S6][S10] | | ● | | ● | | | | P | Proposing an improved term weighting scheme, namely, Developers Preferred Term Frequency/Inverse Document Frequency (DPTF/IDF) |
| Refactoring [S9][S26][S30] | ● | ● | | | | | | P | Solving the problem of missing symbols, misplaced symbols and repeated symbols |
| Syntax Tree [S21] | | ● | | | | | | P | Primary identifier keywords are converted to comment keywords by their similarity in appearance in the syntax tree location |
| Code Comments [S36] | ● | | | | | | | P | Introducing different types of annotations to some extent compensate for vocabulary mismatches between requirements and source code to improve the accuracy of tracing links. |
| Commonality and Variability Analysis（CVA）[S31] | ● | | | | | | | P | Analyzing to which products elements (e.g., requirements, code elements) belong. |
| Term Classification [S11] | | ● | | | | | | P | Categorizing class names, comments, and all other terms in code |
| Analyzing Close Relations [S7] | ● | | | | | | | G | Calculating the close relations (semantic similarity) between target artifacts |
| Verb-object Phrases [S1] | ● | | | | | | | G | Extracting verb-object phrases as main information and essential meaning |
| Model-Driven Engineering (MDE) [S13] | ● | | | | | | | 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 [S15][S23] | ● | | | | ● | | | G | Combing VSM and BTM which can help relieve data sparsity caused by short text |
| Genetic Algorithm [S23] | | | | | ● | | | G | Configuring initial parameters of BTM by introducing Genetic Algorithm |
| Code Calling Relationships [S14] | | | | | | ● | | G | Identifying errors between requirements and code traces by code-calling relationships |
| Historical Co-change Information [S17] | ● | ● | | | | | | G | Taking the processed corpora and co-change information of classes as input to reorder and filter baseline links |
| Configuration Management Log [S29][S31] | ● | | | | | | | G | Restoring links by finding revisions in the configuration management log that contain words related to requirements |
| Adaptive User Feedback [S34] | ● | | | | | | | G | Determining whether and how to apply relevant feedback based on the verbosity of the software artifacts and the number of correct links and false positives that have been categorized. The accuracy is improved by combining the judgment provided by users on classified links. |
| Trace Link Evolver (TLE) [S35] | ● | ● | | | | | | G | Proposing a TLE, which relies on a set of heuristics combined with refactoring detection tools and IR algorithms, to detect predefined change scenarios that occur between successive versions of a software system. |
| Code Ownership [S37] | ● | | | | | | ● | G | Using source code ownership information to capture relationships between source code artifacts to improve recovery of trace links between documents and source code. |
| Mining software repositories [S16] | ● | | ● | | | | | G | An expert is introduced to discard/reorder the baseline traceability links. The expert can submit messages and bugs reports by mining CVS/SVN in the software repository, and store all recovered requirements and links between the software repository in a dedicated set. |
| Dynamic Integration of Structural and Co-change Coupling [S22] | ● | | | | | | | R | Retrieving indirect links based on weighted integration of structural coupling and class coupling based on change history |
| Cluster hypothesis [S24] | ● | | | | | | | R | By discovering appropriate clustering mechanisms, distinguish between high quality and low-quality clusters, filter links in low quality clusters, catch all possible correct traceability links, and reduce false links. |
| Frugal User Feedback with Closeness Analysis on Code [S33] | ● | ● | ● | | | | | R | Introducing only a small amount of user feedback into the closeness analysis on call and data dependencies in code |
| User Feedback [S29] | ● | | | | | | | R | Introducing user validation for candidate links to improve accuracy |
| Analyzing Closeness of Code Dependencies [S5] | ● | ● | ● | | | | | R | Quantifying the interaction degree of call dependency and data dependency between two code classes |
| Class Clustering [S11] | | ● | | | | | | R | The products in the clustering have similar trace relationships |
| Correlation among Classes [S19] | ● | | | | | | | R | Using structural or co-changing dependencies or both to find correlations between classes and use these dependencies to verify traceability links |
| Graph Clustering [S28] | ● | ● | | | | | | R | Information about the cohesion of artifacts within a level of refinement helps improve the trace retrieval process between levels of refinement |
| ConPOS approach [S32] | ● | | ● | | | | | 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 8.** 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 | 18 | http://www.coest.org/ | [S2][S5][S6][S7][S8][S9][S10][S17][S19][S20][S22][S24][S26][S30][S33][S34][S35][S37] |
| | Requirements (50) | Code (299) | 14950 | 314 | | | | |
| | Use Cases (33) | JSP (47) | 1551 | 58 | Small | | | |
| eTour | Use Cases (58) | Code (116) | 6728 | 308 | Large | 13 | http://www.coest.org/ | [S1][S2][S8][S9][S12][S18][S23][S24][S25][S26][S30][S37][S38] |
| | Requirements (58) | Code (116) | 6728 | 366 | | | | |
| EasyClinic | Requirements (30) | Code (47) | 1410 | 83 | Small | 7 | http://www.coest.org/ | [S1][S3][S7][S15][S23][S28][S35] |
| | Use Cases (30) | Test Cases (63) | 1890 | 63 | | | | |
| | UML Interaction Diagram (20) | Code Classes (47) | 940 | 69 | | | | |
| CM-1 | High-level Requirements (235) | Design (220) | 51700 | 361 | Large | 5 | http://www.coest.org/ | [S3][S7][S8][S24][S28] |
| | Requirements (235) | Design (220) | 51700 | 361 | | | | |
| | Requirements (235) | Use Case (Unclear) | Unclear | Unclear | | | | |
| | Requirement (298) | Code (90) | 26820 | 546 | Unclear | | | |
| Pooka | Requirements (41) | Test Cases (25) | 1025 | 51 | Large | 5 | http://www.suberic.net/pooka/ | [S6][S10][S16][S17][S33] |
| | Requirements (90) | Code (298) | 26820 | 546 | | | | |
| EBT | requirements (40) | Code (50) | 2000 | 98 | Small | 4 | http://www.coest.org/ | [S12][S15][S18][S23] |
| | Requirements (16) | Code (124) | 1173 | 315 | | | | |
| GanttProject | Requirements (17) | Code (55) | 935 | 54 | Small | 4 | http://www.ganttproject.biz | [S5][S7][S14][S34] |
| Albergate | Requirements (82) | Code (1771) | 145222 | 871 | Small | 3 | http://www.coest.org/ | [S12][S18][S25] |
| SIP Communicator | Non-functional Requirements (21) | Software Requirements Specification (89) | 1869 | 58 | Large | 3 | http://www.jitsi.org | [S16][S17][S33] |
| WARC | Functional Requirements (43) | Software Requirements Specification (89) | 3827 | 78 | Small | 2 | http://www.coest.org/ | [S15][S23] |
| | High-level Requirements (17) | Low-level Requirements (69) | 1173 | 68 | Large | | | |
| GANNT | Use Cases (67) | Code (100) | 6700 | 1044 | Small | 1 | http://www.coest.org/ | [S7] |
| SMOS | Requirements (34) | Code (483) | 16422 | Unclear | Large | 1 | http://www.coest.org/ | [S38] |
| jEdit v4.3 | Requirements (237) | Code (388) | 91956 | 1515 | Large | 1 | http://www.jedit.org. | [S16] |
| Infinispan | Requirements (116) | Code (413) | 47908 | 744 | Large | 1 | http://infinispan.org/ | [S34] |
| Lucene | Requirements (268) | Code (138) | 36984 | Unclear | Large | 1 | http://lucene.apache.org | [S6] |
| Rhino v1.6 | Requirements (90) | Code (298) | 26820 | 507 | Large | 1 | http://www.mozilla.org/rhino/ | [S16] |
| Lynx | Requirements (128) | Code (unclear) | Unclear | 376 | Unclear | 1 | http://lynx.isc.org/ | [S33] |
| Maven | Requirements (68) | Code (236) | 16048 | 356 | Large | 1 | http://maven.apache.org/ | [S34] |
| Pig | Requirements (Unclear) | Code (Unclear) | Unclear | Unclear | Large | 1 | https://pig.apache.org/ | [S34] |
| Mylyn | Requirements (16) | Code (144) | 2,304 | 221 | Unclear | 1 | http://www.eclipse.org/mylyn/developers | [S20] |
| jHotDraw (JHD) | High-level Requirements (19) | Low-level Requirements (49) | 931 | 567 | Small | 2 | | [S5][S14] |
| MODIS | Requirements (26) | Code (521) | 13,546 | 229 | Small | 2 | | [S3][S35] |
| WDS | requirements (8) | Code (408) | 3240 | 3240 | Large | 2 | | [S26][S30] |
| Chess | Requirements (15) | Code (unclear) | Unclear | Unclear | Large | 1 | | [S14] |
| CUnit | Requirements (19) | Code (61) | 1159 | 104 | | | | |
| iBooks | Use Cases (21) | Code (20) | 420 | 45 | Small | 1 | | [S1] |
| iRobot | Use Cases (24) | Code (14) | 336 | 37 | Small | 1 | | [S4] |
| iTruck | Use Cases (18) | Code (54) | 872 | 51 | Small | 1 | | [S4] |
| iSudoku | Requirements (64) | Code (102) | 6,528 | 1,071 | Small | 1 | | [S4] |
| SMS | Use Cases (24) | Test Cases (60) | 1,440 | 711 | Small | 1 | | [S1] |
| MR0 | Defect Reports (135) | Use Cases (28) | 3,780 | Unclear | Small | 1 | | [S3] |
| MR1 | Change Requests (28) | Use Cases (21) | 588 | Unclear | Small | 1 | | [S3] |
| MR2 | Requirements (49) | Use Cases (51) | 2,499 | Unclear | Small | 1 | | [S3] |
| Pine | Requirements (13) | Code (169) | 2,197 | Unclear | Small | 1 | | [S7] |
| VideoOnDemand (VoD) | Requirements (unclear) | Use Cases (unclear) | Unclear | Unclear | Small | 1 | | [S14] |
| Waterloo | Requirements (88) | Code (208) | 18,304 | Unclear | Unclear | 1 | | [S28] |
| LEDA | Requirements (49) | Code (unclear) | Unclear | Unclear | Large | 1 | | [S25] |
| network control system | Requirements (9) | Code (4) | 36 | 11 | Unclear | 1 | | [S32] |
| Domain Analysis App | Requirements (7) | Code (5) | 35 | 7 | Small | 1 | | [S36] |

## 2. Search process record

| Database | Number of searches | Number of repetitions in each database | Number of each database (After deleting repetitions) | Number of repetitions in all 6 databases | Total number (After deleting repetitions) |
|---|---|---|---|---|---|
| IEEE | 485 | 369 | 116 | 193 | 1618 |
| EI | 1156 | 639 | 517 | | |
| Science Direct | 87 | 47 | 40 | | |
| Springer | 266 | 0 | 266 | | |
| ACM | 62 | 31 | 31 | | |
| Google scholar | 1073 | 232 | 841 | | |

*2.1.* Search records

Search terms:

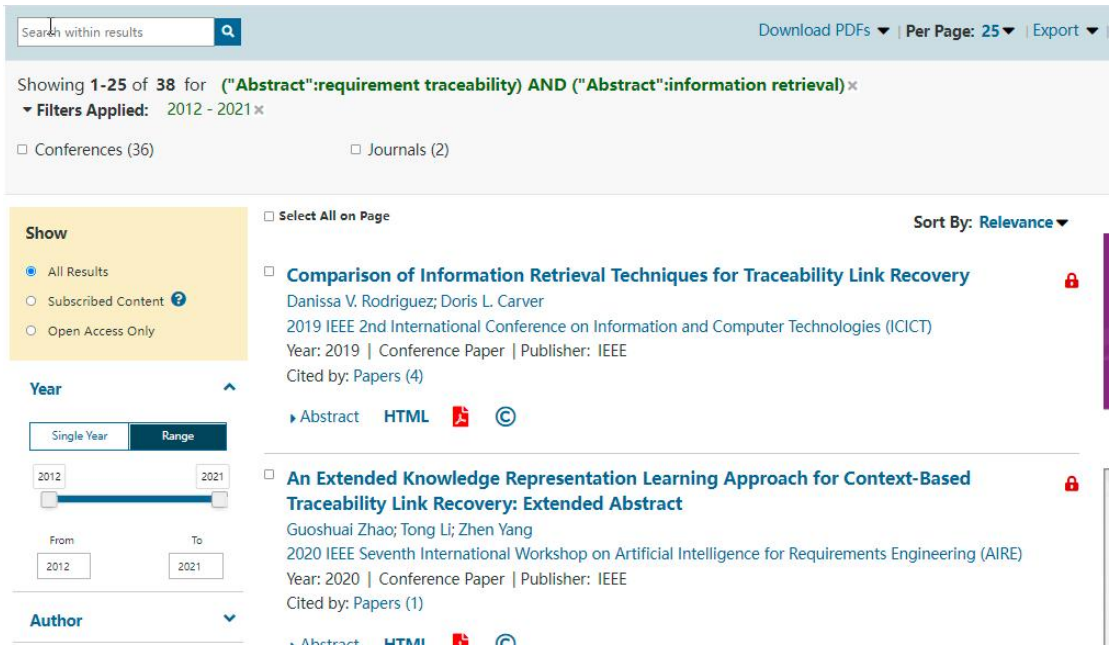
| | |
|---|---|
| P1 | requirement traceability |
| P2 | requirement trace |
| P3 | requirement tracing |
| P4 | requirement traceability recovery |
| I1 | information retrieval |
| I2 | IR |
| I3 | semantic |

(1) IEEE

| | Abstract | Title | Index term |
|---|---|---|---|
| P1+I1 | 38 | 2 | 49 |
| P1+I2 | 21 | 0 | 12 |
| P1+I3 | 38 | 3 | 20 |
| P2+I1 | 34 | 0 | 23 |
| P2+I2 | 17 | 0 | 4 |
| P2+I3 | 27 | 1 | 11 |
| P3+I1 | 34 | 0 | 23 |
| P3+I2 | 17 | 0 | 4 |
| P3+I3 | 27 | 1 | 11 |
| P4+I1 | 13 | 1 | 20 |
| P4+I2 | 9 | 0 | 9 |
| P4+I3 | 9 | 1 | 6 |
| Total | 485 | | |

An example screenshots of search process in IEEE

P1+I1：Abstract

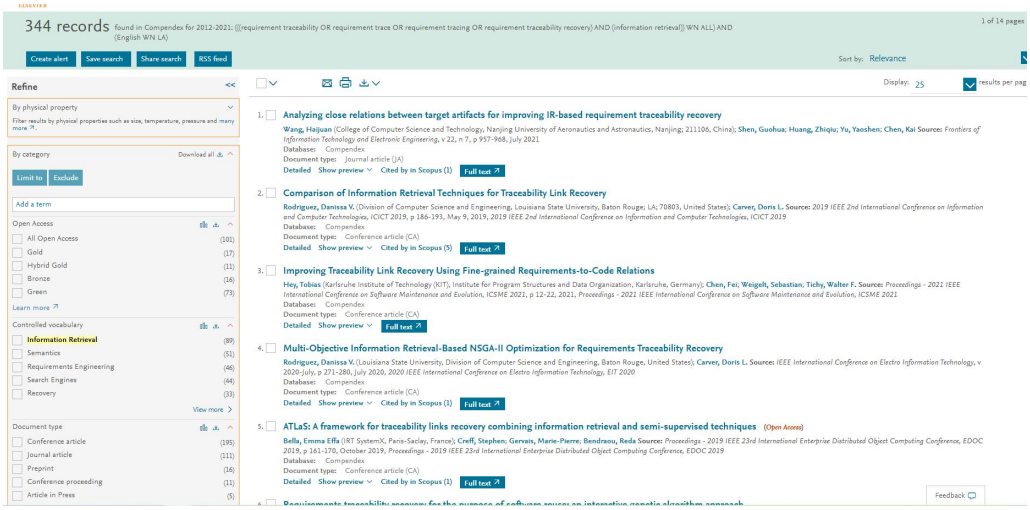



(2) EI

| | Abstract + Title + Keywords (Index term) |
|---|---|
| (p1 or p2 or p3 or p4) and (I1) | 344 |
| (p1 or p2 or p3 or p4) and (I2) | 270 |
| (p1 or p2 or p3 or p4) and (I3) | 542 |
| Total | 1156 |

The screenshots of search process in EI

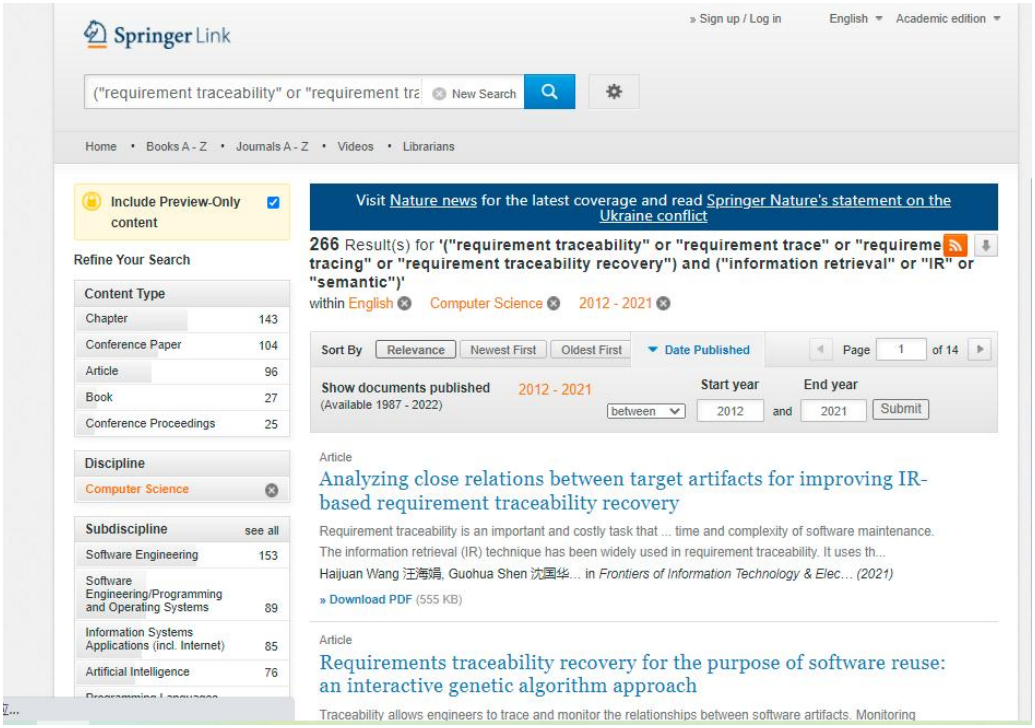
(p1 or p2 or p3 or p4) and (I1): Abstract + Title + Keywords




(3) Springer

| | Abstract + Title + Keywords (Index term) |
|---|---|
| (p1 or p2 or p3 or p4) and (I1 or I2 or I3) | 266 |

The screenshots of search process in Springer:

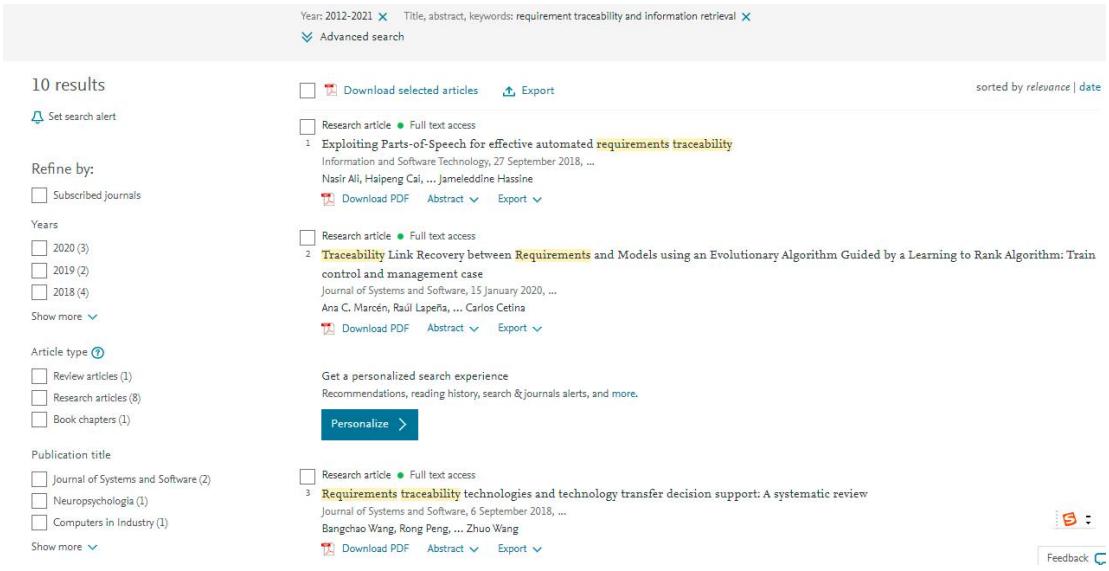



(4) Science Direct

| | Abstract + Title + Keywords (Index term) |
|---|---|
| P1+I1 | 10 |
| P1+I2 | 16 |
| P1+I3 | 12 |
| P2+I1 | 10 |
| P2+I2 | 16 |
| P2+I3 | 12 |
| P3+I1 | 3 |
| P3+I2 | 4 |
| P3+I3 | 3 |
| P4+I1 | 1 |
| P4+I2 | 0 |
| P4+I3 | 0 |
| Total | 87 |

An example screenshots of search process in Elsevier

P1+ I1： Abstract + Title + Keywords：

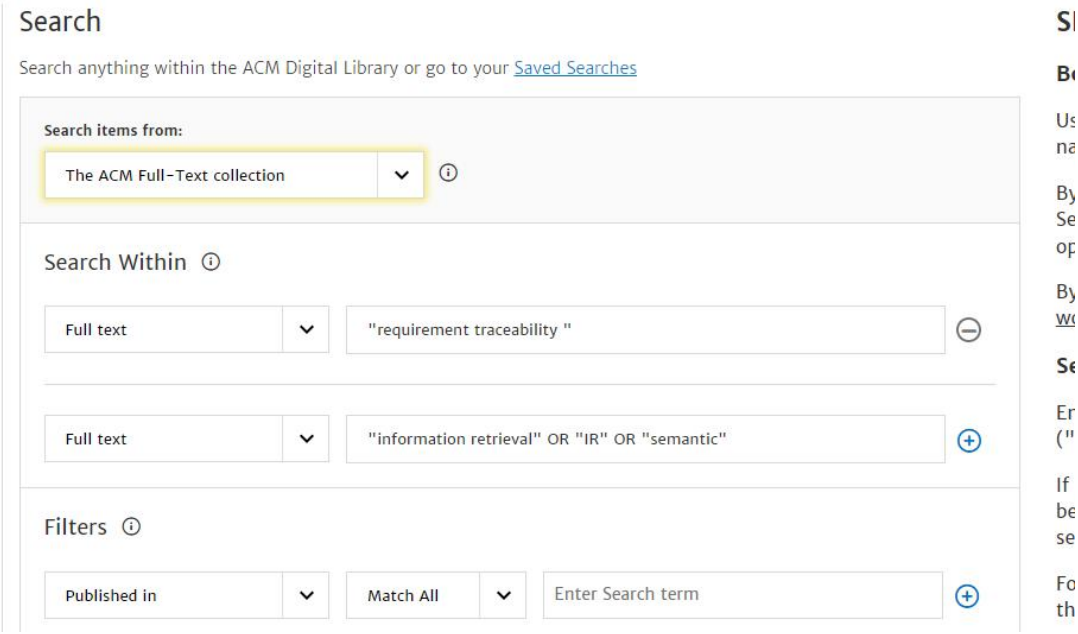


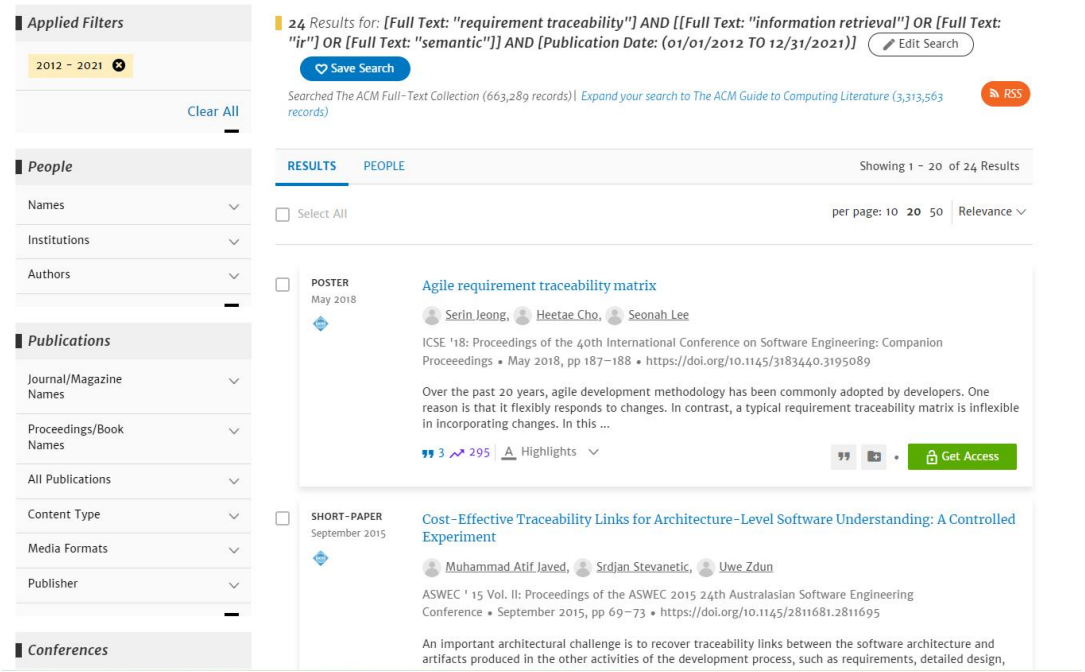

(5) ACM

| | Full text | Anywhere |
|---|---|---|
| P1+(I1, I2, I3) | 24 | 24 |
| P2+(I1, I2, I3) | 4 | 4 |
| P3+(I1, I2, I3) | 3 | 3 |
| P4+(I1, I2, I3) | 0 | 0 |
| Total | 62 | |

An example screenshots of search process in ACM

Full text：






(6) Google Scholar

| | Title | Abstract | Index term |
|---|---|---|---|
| (p1 or p2 or p3 or p4) and (I1 or I2 or I3) | 216 | 7 | 850 |
| Total | 1073 | | |

An example screenshots of search process in Google Scholar

intext：