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 Wan Chengcheng Jin Bo	2016	SNPD	Conference
S2	An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis	Jiale Zhou Yue Lu Kristina Lundqvist	2013	Semantic Scholar	Other
S3	Configuring Latent Semantic Indexing for Requirements Tracing	Sebastian Eder Henning Femmer Benedikt Hauptmann 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 Yue Lu Kristina Lundqvist	2013	SEAA	Conference
S5	Analyzing closeness of code dependencies for improving IR-based Traceability Recovery	Hongyu Kuang Jia Nie Hao Hu Patrick Rempel Jian Lü Alexander Egyed Patrick Mäder	2017	SANER	Conference
S6	An empirical study on the importance of source code entities for requirements traceability	Nasir Ali Zohreh Sharafi Yann-Gaël Guéhéneuc Giuliano Antoniol	2014	ESE	Journal
S7	Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery	Haijuan Wang Guohua Shen Zhiqiu Huang Yaoshen Yu Kai Chen	2021	FITEE	Journal
S8	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	Wiem Khlif Dhikra Kchaou Nadia Bouassida	2022	IJCAI	Journal
S9	Achieving better requirements to code traceability： which refactoring should be done first?	Farina Faiz Rubaida Easmin Alim Ul Gias	2016	QUATIC	Conference
S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	Hui Gao Hongyu Kuang Kexin Sun Xiaoxing Ma Alexander Egyed Patrick Mäder Guoping Rong Dong Shao He Zhan	2022	ASE	Conference
S11	An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing	Jianwei Shao Wei Wu Peng Geng	2013	ICCSA	Conference
S12	An IR-based Artificial Bee Colony Approach for Traceability Link Recovery	Danissa V. Rodriguez Doris L. Carver	2020	ICTAI	Conference

S13	Propagating frugal user feedback through closeness of code dependencies to improve IR-based traceability recovery	Hui Gao Hongyu Kuang Xiaoxing Ma Hao Hu Jian Lü Patrick Mäder Alexander Egyed	2022	ESE	Journal
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	Raúl Lapeña Francisca Pérez Carlos Cetina Óscar Pastor	2022	RE Conference	Conference
S15	Combining VSM and BTM to improve requirements trace links generation	Bangchao Wang Rong Peng Zhuo Wang Yaxin Zhao	2019	SEKE	Conference
S16	Trustrac： Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links	Nasir Ali Yann-Gaël Guéhéneuc Giuliano Antoniol	2013	TSE	Journal
S17	Leveraging historical co-change information for requirements traceability	Nasir Ali Fehmi Jaafar Ahmed E. Hassan	2013	WCRE	Conference
S18	Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery	Danissa V. Rodriguez Doris L. Carver	2020	EIT	Conference
S19	Filtering of false positives from IR-based traceability links among software artifacts	Jyoti Jitender Kumar Chhabra	2017	I2CT	Conference
S20	Quality improvements for trace links between source code and requirements	Paul Hübner	2016	REFSQ	Conference
S21	Evaluation of Natural Language Processing for Requirements Traceability	Christopher D. Laliberte Ronald E. Giachetti Mathias Kolsch	2022	SOSE	Conference
S22	Requirements Traceability Through Information Retrieval Using Dynamic Integration of Structural and Co-change Coupling	Jyoti Jitender Kumar Chhabra	2017	ICAICR	Journal
S23	An Automated Hybrid Approach for Generating Requirements Trace Links	Wang Bangchao Peng Rong Wang Zhuo Wang Xiaomin Li Yuanbang	2020	IJSEKE	Journal
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	Manpreet Kaur Harpreet Kaur	2021	ICCMST	Conference
S25	Search-Based Requirements Traceability Recovery： A Multi-Objective Approach	Adnane Ghannem Mohamed Salah Hamdi Marouane Kessentini Hany H. Ammar	2017	CEC	Conference and Workshop Papers
S26	Supporting requirements traceability through refactoring	Anas Mahmoud 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 Smita Ghaisas Ashish Sureka	2014	RAISE	Conference
S28	Towards feature-aware retrieval of refinement traces	Patrick Rempel Patrick Mäder Tobias Kuschke	2013	TEFSE	Conference
S29	Interactive recovery of requirements traceability links using user feedback and configuration management logs	Ryosuke Tsuchiya Hironor Washizakii Yoshiaki Fukazawa Keishi Oshima Ryota Mibe	2015	CAiSE	Conference
S30	Supporting requirements to code traceability through refactoring	Anas Mahmoud Nan Niu	2014	RE Conference	Journal

S31	Recovering traceability links between requirements and source code using the configuration management log	Ryosuke Tsuchiya Hironori Washizaki Yoshiaki Fukazawa Tadahisa Kato Masumi Kawakami Kentaro Yoshimura	2015	IEICE	Journal
S32	Exploiting Parts-of-Speech for Effective Automated Requirements Traceability	Nasir Ali Haipeng Cai Abdelwahab Hamou-Lhadj Jameleddine Hassine	2018	IST	Journal
S33	Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery	Hongyu Kuang Hui Gao Hao Hu Xiaoxing Ma Jian Lü Patrick Mäder Alexander Egyed	2019	ICPC	Conference
S34	Adaptive User Feedback for IR-Based Traceability Recovery	Annibale Panichella Andrea De Lucia Andy Zaidman	2015	SST	Conference
S35	Evolving Software Trace Links between Requirements and Source Code	Mona Rahimi Jane Cleland-Huang	2018	ESE	Journal
S36	Supporting Requirements to Code Traceability Creation by Code Comments	Guohua Shen Haijuan Wang Zhiqiu Huang YaoShen Yu Kai Chen	2021	IJSEKE	Journal
S37	Using code ownership to improve IR-based Traceability Link Recovery	Diana Diaz Gabriele Bavota Andrian Marcus Rocco Oliveto Silvia Takahashi 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	Preprocessing 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 LSI JS	Links Refinement Stage
S6	An empirical study on the importance of source code entities for requirements traceability	LSI TM(LDA)	Preprocessing Stage
S7	Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery	VSM	Links Generation Stage
S8	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	LSI	Preprocessing Stage
S9	Achieving better requirements to code traceability： which refactoring should be done first?	VSM LSI	Preprocessing Stage
S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	LSI VSM JS	Links Refinement Stage 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 Links Refinement Stage
S12	An IR-based Artificial Bee Colony Approach for Traceability Link Recovery	IR-based+	Links Generation Stage
S13	Propagating frugal user feedback through closeness of code dependencies to improve IR-based traceability recovery	VSM LSI JS	Links Refinement Stage Preprocessing Stage
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	LSI	Links Generation Stage
S15	Combining VSM and BTM to improve requirements trace links generation	VSM TM(BTM)	Links Generation Stage
S16	Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links	VSM JS	Links Generation Stage
S17	Leveraging historical co-change information for requirements traceability	VSM 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	Evaluation of Natural Language Processing for Requirements Traceability	VSM	None
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 TM(BTM)	Links Generation Stage
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	VSM LSI JS	None
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）	None
S28	Towards feature-aware retrieval of refinement traces	VSM	Links Refinement Stage
S29	Interactive recovery of requirements traceability links using user feedback and configuration management logs	VSM	Links Generation Stage Links Refinement Stage
S30	Supporting requirements to code traceability through refactoring	VSM LSI	Preprocessing Stage
S31	Recovering traceability links between requirements and source code using the configuration management log	VSM	Preprocessing Stage Links Generation Stage Links Refinement Stage
S32	Exploiting Parts-of-Speech for Effective Automated Requirements Traceability	VSM JS	Links Refinement Stage
S33	Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery	VSM LSI 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 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 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	Heuristic Measures
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 Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	Traceability Rules
S9	Achieving better requirements to code traceability： which refactoring should be done first?	Refactoring
S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	Consensual Biterms Global and Local Weight
S11	An Improved Approach to the Recovery of Traceability Links between Requirement Documents and Source Codes Based on Latent Semantic Indexing	Term Classification Class Clustering
S12	An IR-based Artificial Bee Colony Approach for Traceability Link Recovery	Artificial Bee Colony (ABC) Algorithm
S13	Propagating frugal user feedback through closeness of code dependencies to improve IR-based traceability recovery	Frugal User Feedback with Closeness Analysis on Code Analyzing Closeness of Code Dependencies
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	BPMN-specific approaches
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	Non-dominated Sorting Genetic Algorithm (NSGA-II)
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	Non-dominated Sorting Genetic Algorithm (NSGA-II)
S21	Evaluation of Natural Language Processing for Requirements Traceability	None
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 Genetic Algorithm
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	None
S25	Search-Based Requirements Traceability Recovery： A Multi-Objective Approach	Non-dominated Sorting Genetic Algorithm (NSGA-II)
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 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 Commonality and Variability Analysis（CVA） 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 iBooks SMS EasyClinic
S2	An Improved VSM-based Post-Requirements Traceability Recovery Approach Using Context Analysis	Use Cases	Code	eTour iTrust
S3	Configuring Latent Semantic Indexing for Requirements Tracing	Requirements Defect Reports Use Cases Change Requests	Requirements Use Cases Test Cases Use Cases	MODIS CM-1 EasyClinic MR0 MR1 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 iTruck iSudoku
S5	Analyzing closeness of code dependencies for improving IR-based Traceability Recovery	Requirements	Code	iTrust GanttProject jHotDraw
S6	An empirical study on the importance of source code entities for requirements traceability	Requirements	Code	iTrust Lucene Pooka
S7	Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery	Use Case Requirements Requirements Requirements Use Case	Test Case Design Use Case Requirements Code	EasyClinic CM1-NASA Pine GANNT iTrust
S8	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	Use Cases	Code	Car rental Customer Relationships system
S9	Achieving better requirements to code traceability：which refactoring should be done first?	Use Cases	Code	iTrust eTour
S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	Requirements	Code	iTrust GanttProject Maven Pig Infinispan Seam2 Drools Derby Groovy
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 Use Cases	Code	EBT Albergate eTour
S13	Propagating frugal user feedback through closeness of code dependencies to improve IR‑based traceability recovery	Requirements	Requirements	iTrust GanttProject Maven Pig8 Infinispan

				Drools Derby Seam Groovy
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	BPMN models	Requirements	Industrial case study Academic case study
S15	Combining VSM and BTM to improve requirements trace links generation	Use case Requirements Requirements	Test Case Test Case Requirements	WARC EasyClinic EBT
S16	Trustrac：Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links	Requirements	Code	jEdit Pooka Rhino SIP Communicator
S17	Leveraging historical co-change information for requirements traceability	Requirements	Code	iTrust Pooka SIP Communicator
S18	Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery	Requirements	Code	EBT Albergate 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 Use Cases	Code	Mylyn iTrust
S21	Evaluation of Natural Language Processing for Requirements Traceability	Requirements	Requirements	National Aeronautics and Space Administration (NASA)
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 Requirements Requirements Use Cases	Test Case Test Case Requirements Code	WARC subset 1 WARC subset 2 EBT subset EasyClinic eTour
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	Requirements	Code	Activemq Cassandra Derby Hive Mina Pig Solr Synapse Tika Xerces2j
S25	Search-Based Requirements Traceability Recovery：A Multi-Objective Approach	Requirements	Code	LEDA Albergate ETOUR
S26	Supporting requirements traceability through refactoring	Requirements	Code	iTrust eTour 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 Use Case	Use Case Test Case	EasyClinic CM-1 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 eTour WDS
S31	Recovering traceability links between requirements and source code using the configuration management log	Requirements	Code	CUnit Network Control System
S32	Exploiting Parts-of-Speech for Effective Automated Requirements Traceability	Requirements	Code	iTrust Lynx Pooka SIP Communicator
S33	Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery	Requirements	Code	iTrust Maven Pig GanttProject Infinispan
S34	Adaptive User Feedback for IR-Based Traceability Recovery	Use Cases Test Cases Code UML Interaction Diagram Requirements	Code Code JSP Code Requirements	Easy-Clinic i-Trust Modis
S35	Evolving Software Trace Links between Requirements and Source Code	Requirements	Code	Domain Analysis App DOTS File Generator Apache Cassandra Database System Dronology system
S36	Supporting Requirements to Code Traceability Creation by Code Comments	Use Cases	Code	eTour iTrust
S37	Using code ownership to improve IR-based Traceability Link Recovery	Use Cases	Code	eTour SMOS

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

Index	Title	Intercept points	Measure	Evidence level
S1	An empirical study on recovering requirement-to-code links	Threshold	Recall Precision F-Measure	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	Not write	MAP Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S3	Configuring Latent Semantic Indexing for Requirements Tracing	Not write	MAP AP	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	Not write	Recall, Precision, MAP	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 Precision F-Measure, MAP AP	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S6	An empirical study on the importance of source code entities for requirements traceability	Threshold	Recall Precision F-Measure	Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting).
S7	Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery	Threshold	Recall Precision MAP AP	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S8	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	Threshold	Recall Precision	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	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).

S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	Threshold	Precision Recall F- Measure AP MAP	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 Precision	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 Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S13	Propagating frugal user feedback through closeness of code dependencies to improve IR-based traceability recovery	Threshold	Recall Precision F-Measure AP MAP	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	Not write	Recall Precision F-Measure MCC AUC	Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting).
S15	Combining VSM and BTM to improve requirements trace links generation	Threshold	Recall Precision	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 Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S17	Leveraging historical co-change information for requirements traceability	Threshold	Recall Precision	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 Precision F-Measure	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 Precision F-Measure	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 Precision	Level 1. Evidence obtained from demonstration or working out with toy examples.
S21	Evaluation of Natural Language Processing for Requirements Traceability	Threshold	Recall Precision F-Measure	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, Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S23	An Automated Hybrid Approach for Generating Requirements Trace Links	Selectivity	Recall Precision F-Measure Selectivity	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	Not write	Recall Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S25	Search-Based Requirements Traceability Recovery： A Multi-Objective Approach	Iteration	Recall Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S26	Supporting requirements traceability through refactoring	Threshold	Recall Precision MAP DiffAR	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	Not write	AP MAP	Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting).
S28	Towards feature-aware retrieval of refinement traces	Not write	Recall Precision AP	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 Precision F-Measure	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 Precision MAP DiffAR	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 Precision F-Measure	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	Not write	Recall Precision MAP AP	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 Precision F-Measure MAP AP	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S34	Adaptive User Feedback for IR-Based Traceability Recovery	Threshold	Recall Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S35	Evolving Software Trace Links between Requirements and Source Code	Not write	Recall Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
S36	Supporting Requirements to Code Traceability Creation by Code Comments	Not write	Recall Precision MAP AP	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 Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).

Table 1. 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		
Verb-object Phrases [S1]	●							P	Extracting verb-object phrases as main information and essential meaning.
Context- based [S2][S4]	●							P	Separating intent from context in requirements.
Improved Term Weighting Scheme [S6]		●		●				P	Proposing an improved term weighting scheme, namely, Developers Preferred Term Frequency/Inverse Document Frequency (DPTF/IDF) .
Traceability Rules [S8]		●						P	Defining traceability rules to determine correspondences between the requirement modeled with the use case diagram based on the enriched textual description and design diagrams modeled
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 comments 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.
Consensual Biterms [S10]	●	●	●						Extracting consensual biterms to first enrich the corpus for IR techniques.
Heuristic Measures [S3]		●						G	A fully automated technique to determine appropriate configurations for LSI to recover links between requirements artifacts.
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
BPMN-specific approaches [S14]		●						G	An approach that improves the results of TLR between requirements and BPMN models.
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.
Analyzing Close Relations [S7]	●							R	Calculating the close relations (semantic similarity) between target artifacts
Global and Local Weight[S10]	●	●	●					R	Using consensual biterms to adjust global and local weight to adjust the ranking of candidate lists.
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][S13]	●	●	●					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][S13]	●	●	●					P,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.
Classification [S31]	●							R	Classifying traceability links into 5 five types using the CVA results, then using the classification to refine links.

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

Table 2. Dataset's 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	17	http://www.coest.org/	[S2][S5][S6][S7][S10][S9][S13][S17][S19][S20][S22][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	11	http://www.coest.org/	[S1][S2][S9][S12][S18][S23][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				
GanttProject	Requirements (17)	Code (55)	935	54	Small	6	http://www.ganttproject.biz	[S5][S7][S14][S34][S10][S13]
	Requirements (16)	Code (124)	1984	315			https://github.com/bardsoftware/ganttproject	
Pooka	Requirements (41)	Test Cases (25)	1025	51	Large	4	http://www.suberic.net/pooka/	[S6][][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				
Pig	Requirements (Unclear)	Code (Unclear)	Unclear	Unclear	Large	4	https://pig.apache.org/	[S34] [S10] [S13] [S24]
	Requirements (87)	Code (289)	25143	547			https://github.com/apache/pig	
	Requirements (58)	Code (754)		Unclear				
CM-1	High-level Requirements (235)	Design (220)	51700	361	Large	3	http://www.coest.org/	[S3][S7][S28]
	Requirements (235)	Use Case (Unclear)	Unclear	Unclear				
	Requirement (298)	Code (90)	26820	546				
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]
Derby	Requirements (390)	Code (611)	238290	2315	Large	3	https://github.com/apache/derby	[S10] [S13] [S24]
	Requirements (133)	Code (2184)		Unclear	Large			
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			
Infinispan	Requirements (116)	Code (413)	47908	744	Large	3	http://infinispan.org/	[S34]
	Requirements (232)	Code (319)	74008	1116	Large		https://github.com/infinispan/infinispan	[S10] [S13]
Maven	Requirements (68)	Code (236)	16048	356	Large	3	http://maven.apache.org/	[S34]
	Requirements (36)	Code (82)	2880	151			https://github.com/apache/maven	[S10] [S13]
Seam2	Requirements (189)	Code (150)	28350	463	Large	2	http://www.seamframework.org/Seam2.html	[S10] [S13]
Drools	Requirements (183)	Code (248)	45384	841	Large	2	https://github.com/kiegroup/drools	[S10] [S13]
Groovy	Requirements (104)	Code (100)	10400	180	Large	2	https://github.com/apache/groovy	[S10] [S13]
MODIS	Requirements (26)	Code (521)	13,546	229	Small	2		[S3] [S35]
WDS	Requirements (8)	Code (408)	3240	3240	Large	2		[S26] [S30]
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]
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]
Mylyn	Requirements (16)	Code (144)	2,304	221	Unclear	1	http://www.eclipse.org/mylyn/developers	[S20]
Lynx	Requirements (128)	Code (unclear)	Unclear	376	Unclear	1	http://lynx.isc.org/	[S33]
jHotDraw (JHD)	High-level Requirements (19)	Low-level Requirements (49)	931	567	Small	1		[S5]
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]
Waterloo	Requirements (88)	Code (208)	18,304	Unclear	Unclear	1		[S28]
LEDA	Requirements (49)	Code (unclear)	Unclear	Unclear	Unclear	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]
Car rental system	Use cases (9)	Code (98)	882	Unclear	Small	1		[S8]
Customer Relationships system	Use cases (7)	Code (65)	455	Unclear	Small	1		[S8]
ActiveMQ	Requirements (93)	Code (2761)		Unclear	Large	1		[S24]
Cassandra	Requirements (65)	Code (328)		Unclear	Large	1		[S24]
Hive	Requirements (92)	Code (732)		Unclear	Large	1		[S24]
Mina	Requirements (48)	Code (272)		Unclear	Large	1		[S24]
Solr	Requirements (59)	Code (74)		Unclear	Large	1		[S24]
Synapse	Requirements (44)	Code (877)		Unclear	Large	1		[S24]
Tika	Requirements (45)	Code (78)		Unclear	Large	1		[S24]
Xerces2j	Requirements (26)	Code (705)		Unclear	Large	1		[S24]
the National Aeronautics and Space Administration (NASA)	Requirements	Requirement	Unclear	Unclear	Unclear	1		[S21]

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	476	359	117	193	1618
EI	1010	493	517		
Science Direct	39	0	39		
Springer	408	142	266		
ACM	68	37	31		
Google scholar	1198	357	841		

2.1. Search records

Search terms:

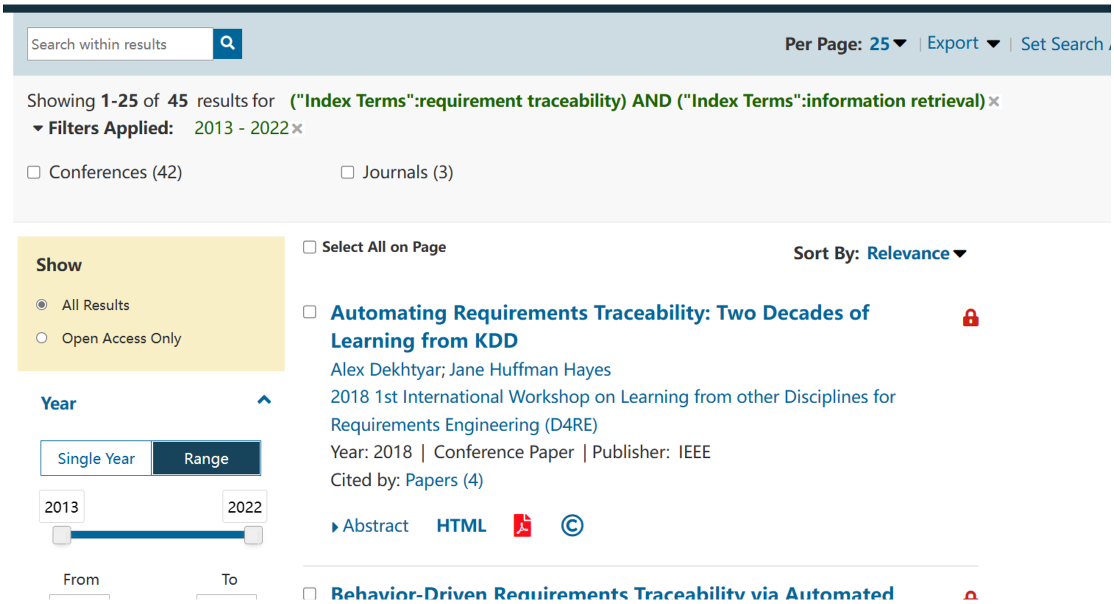
P1	requirement traceability
P2	requirement trace
P3	requirement tracing
P4	requirement traceability recovery
I1	information retrieval
I2	IR
I3	semantic

(1) IEEE

	Abstract	Title	Index term
P1+I1	34	1	45
P1+I2	20	1	11
P1+I3	40	3	21
P2+I1	30	3	24
P2+I2	16	0	4
P2+I3	28	1	13
P3+I1	30	0	24
P3+I2	16	0	4
P3+I3	28	1	12
P4+I1	12	1	20
P4+I2	9	0	8
P4+I3	9	1	6
Total	476		

An example screenshots of search process in IEEE

P1+I1：Abstract



Showing 1-25 of 45 results for ("Index Terms":requirement traceability) AND ("Index Terms":information retrieval)

Filters Applied: 2013 - 2022

□ Conferences (42) □ Journals (3)

2013 2022

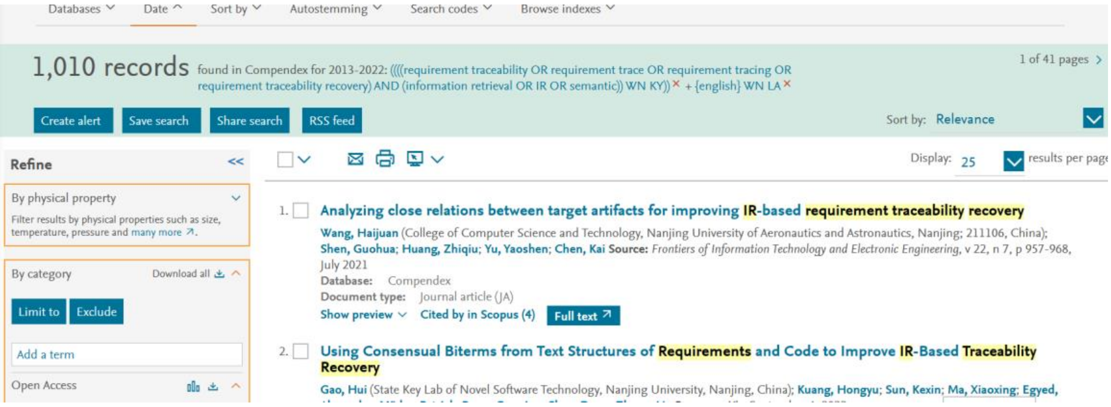
□ **Automating Requirements Traceability: Two Decades of Learning from KDD**
Alex Dekhtyar; Jane Huffman Hayes
2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE)
Year: 2018 | Conference Paper | Publisher: IEEE
Cited by: Papers (4)

□ **Behavior-Driven Requirements Traceability via Automated**

(2) EI

	Abstract + Title + Keywords (Index term)
(p1 or p2 or p3 or p4) and (I1 or I2 or I3)	1010

The screenshots of search process in EI



1,010 records found in Compendex for 2013-2022: ((((requirement traceability OR requirement trace OR requirement tracing OR requirement traceability recovery) AND (information retrieval OR IR OR semantic)) WN KY)) + {english} WN LA

1 of 41 pages

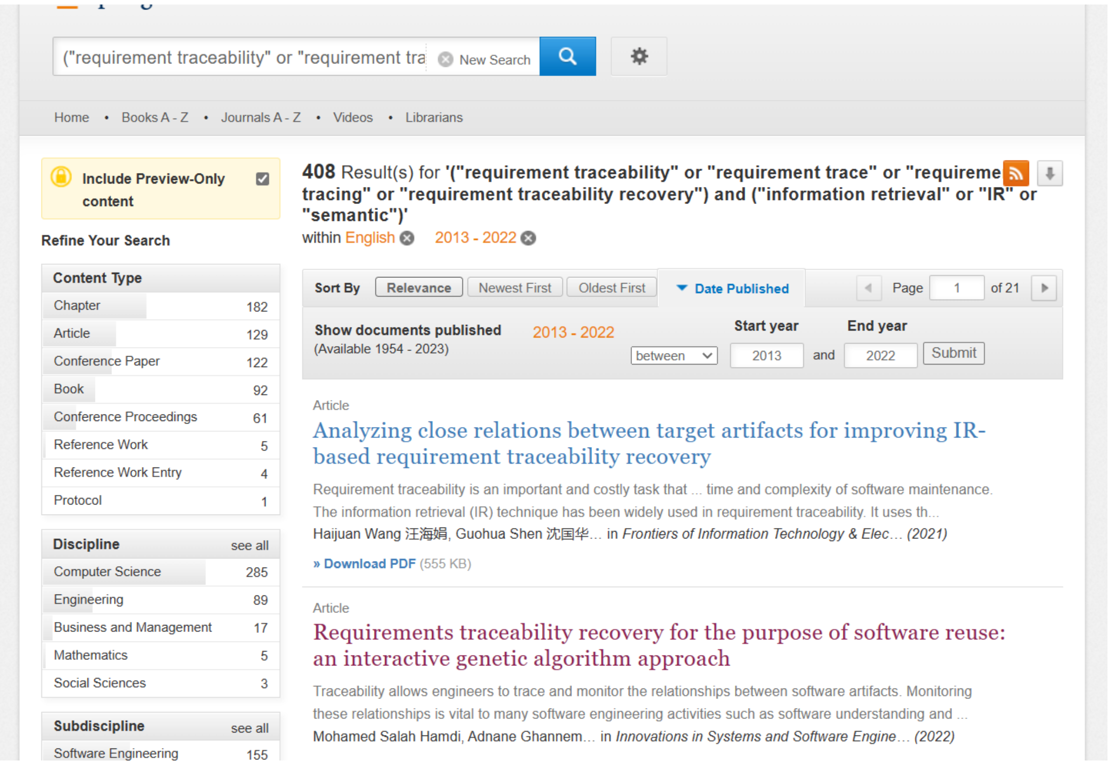
1. □ **Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery**
Wang, Haijuan (College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing; 211106, China); Shen, Guohua; Huang, Zhiqiu; Yu, Yaoshen; Chen, Kai **Source:** *Frontiers of Information Technology and Electronic Engineering*, v 22, n 7, p 957-968, July 2021
Database: Compendex
Document type: Journal article (JA)
Cited by in Scopus (4)

2. □ **Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery**
Gao, Hui (State Key Lab of Novel Software Technology, Nanjing University, Nanjing, China); Kuang, Hongyu; Sun, Kexin; Ma, Xiaoxing; Egyed,

(3) Springer

	Abstract + Title + Keywords (Index term)
(p1 or p2 or p3 or p4) and (I1 or I2 or I3)	408

The screenshots of search process in Springer:



("requirement traceability" or "requirement tra ⊗ New Search

Home • Books A - Z • Journals A - Z • Videos • Librarians

Include Preview-Only content ☑

Refine Your Search

Content Type	
Chapter	182
Article	129
Conference Paper	122
Book	92
Conference Proceedings	61
Reference Work	5
Reference Work Entry	4
Protocol	1

Discipline	see all
Computer Science	285
Engineering	89
Business and Management	17
Mathematics	5
Social Sciences	3

Subdiscipline	see all
Software Engineering	155

408 Result(s) for **'("requirement traceability" or "requirement trace" or "requireme tracing" or "requirement traceability recovery") and ("information retrieval" or "IR" or "semantic")'**
within English ⊗ 2013 - 2022 ⊗

Sort By: Relevance | Newest First | Oldest First | Date Published — Page 1 of 21

Show documents published 2013 - 2022 (Available 1954 - 2023)
between 2013 and 2022 (Start year, End year) Submit

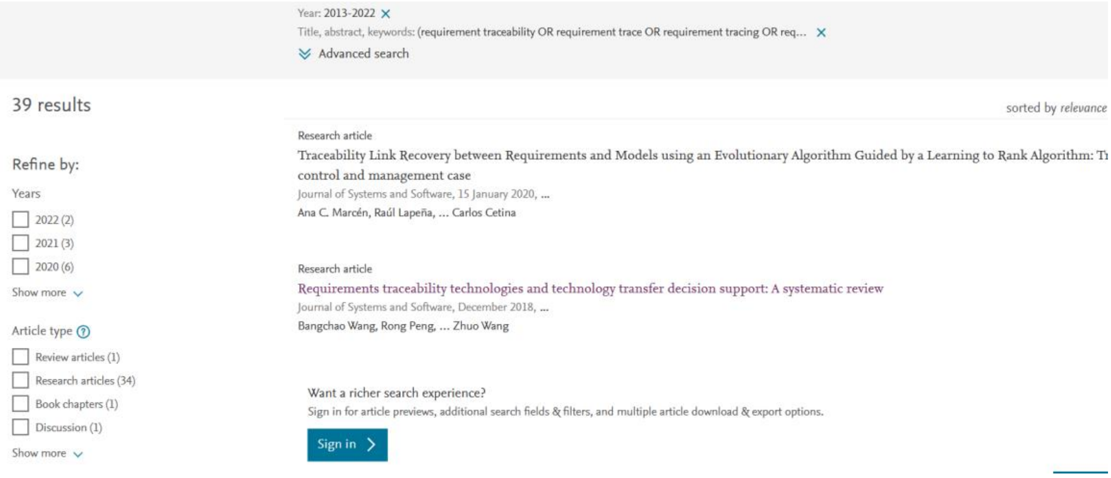
Article
Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery
Requirement traceability is an important and costly task that ... time and complexity of software maintenance. The information retrieval (IR) technique has been widely used in requirement traceability. It uses th...
Haijuan Wang 汪海娟, Guohua Shen 沈国华... in *Frontiers of Information Technology & Elec... (2021)*
» Download PDF (555 KB)

Article
Requirements traceability recovery for the purpose of software reuse: an interactive genetic algorithm approach
Traceability allows engineers to trace and monitor the relationships between software artifacts. Monitoring these relationships is vital to many software engineering activities such as software understanding and ...
Mohamed Salah Hamdi, Adnane Ghannem... in *Innovations in Systems and Software Engine... (2022)*

(4) Science Direct

	Abstract + Title + Keywords (Index term)
(p1 or p2 or p3 or p4) and (I1 or I2 or I3)	39

An example screenshots of search process in Elsevier



Year: 2013-2022 ✕
Title, abstract, keywords: (requirement traceability OR requirement trace OR requirement tracing OR req... ✕
Advanced search

39 results

sorted by *relevance*

Refine by:

Years
□ 2022 (2)
□ 2021 (3)
□ 2020 (6)
Show more

Article type
□ Review articles (1)
□ Research articles (34)
□ Book chapters (1)
□ Discussion (1)
Show more

Research article
Traceability Link Recovery between Requirements and Models using an Evolutionary Algorithm Guided by a Learning to Rank Algorithm: Tr control and management case
Journal of Systems and Software, 15 January 2020, ...
Ana C. Marcén, Raúl Lapeña, ... Carlos Cetina

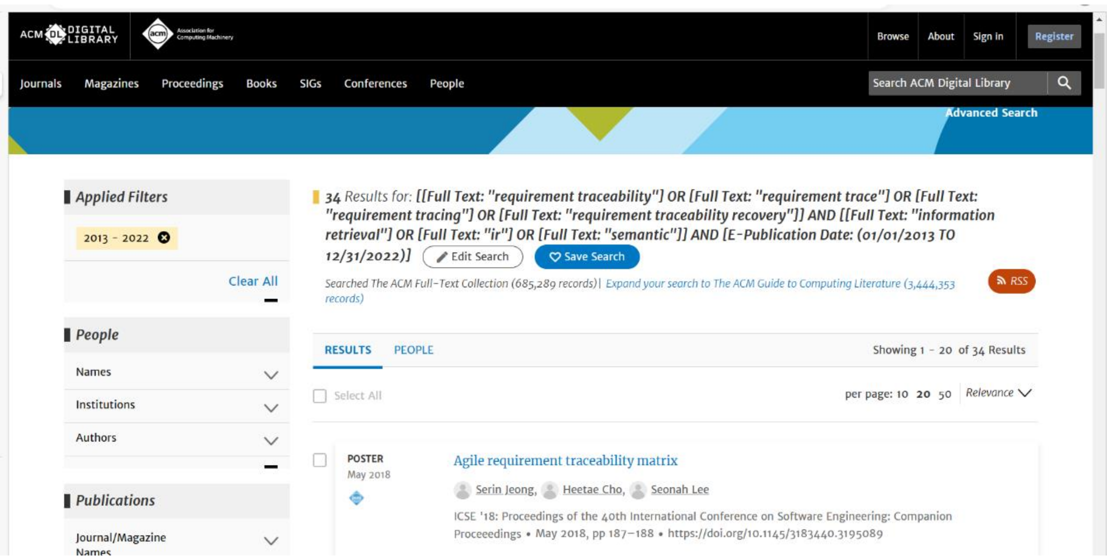
Research article
Requirements traceability technologies and technology transfer decision support: A systematic review
Journal of Systems and Software, December 2018, ...
Bangchao Wang, Rong Peng, ... Zhuo Wang

Want a richer search experience?
Sign in for article previews, additional search fields & filters, and multiple article download & export options.
Sign in

(5) ACM

	Full text	Anywhere
(p1 or p2 or p3 or p4) and (I1 or I2 or I3)	34	34
Total	68	

An example screenshots of search process in ACM
Full text：



34 Results for: [[Full Text: "requirement traceability"] OR [Full Text: "requirement trace"] OR [Full Text: "requirement tracing"] OR [Full Text: "requirement traceability recovery"]] AND [[Full Text: "information retrieval"] OR [Full Text: "ir"] OR [Full Text: "semantic"]] AND [E-Publication Date: (01/01/2013 TO 12/31/2022)]

Searched The ACM Full-Text Collection (685,289 records) | Expand your search to The ACM Guide to Computing Literature (3,444,353 records)

Showing 1 - 20 of 34 Results

per page: 10 20 50

POSTER
May 2018

Agile requirement traceability matrix

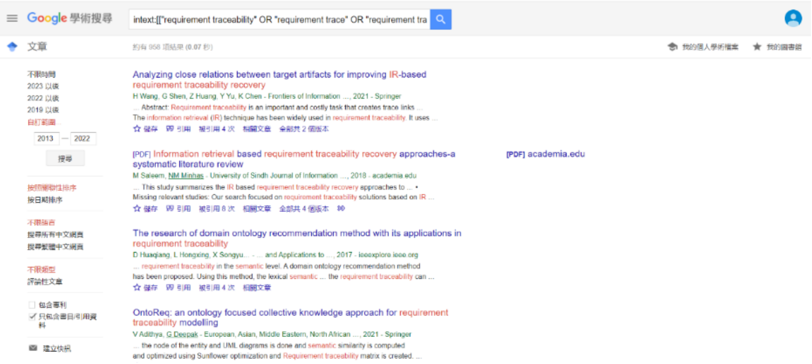
Serin Jeong, Heetae Cho, Seonah Lee

ICSE '18: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings • May 2018, pp 187–188 • https://doi.org/10.1145/3183440.3195089

(6) Google Scholar

	Title	Abstract	Full text
(p1 or p2 or p3 or p4) and (I1 or I2 or I3)	233	7	958
Total	1198		

An example screenshots of search process in Google Scholar
intext：



intext:[["requirement traceability" OR "requirement trace" OR "requirement tra

約有 958 項結果 (0.07 秒)

Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery
H Wang, G Shen, Z Huang, Y Yu, K Chen - Frontiers of Information ..., 2021 - Springer
... Abstract: Requirement traceability is an important and costly task that creates trace links ... The information retrieval (IR) technique has been widely used in requirement traceability. It uses ...
☆ 儲存 99 引用 被引用 4 次 相關文章 全部共 2 個版本

[PDF] Information retrieval based requirement traceability recovery approaches-a systematic literature review
[PDF] academia.edu
M Saleem, NM Minhas - University of Sindh Journal of Information ..., 2018 - academia.edu
... This study summarizes the IR based requirement traceability recovery approaches to ... • Missing relevant studies: Our search focused on requirement traceability solutions based on IR ...
☆ 儲存 99 引用 被引用 8 次 相關文章 全部共 4 個版本

The research of domain ontology recommendation method with its applications in requirement traceability
D Huaqiang, L Hongxing, X Songyu... - ... and Applications to ..., 2017 - ieeexplore.ieee.org
... requirement traceability in the semantic level. A domain ontology recommendation method has been proposed. Using this method, the lexical semantic ... the requirement traceability can ...
☆ 儲存 99 引用 被引用 4 次 相關文章

OntoReq: an ontology focused collective knowledge approach for requirement traceability modelling
V Adithya, G Deepak - European, Asian, Middle Eastern, North African ..., 2021 - Springer
... the node of the entity and UML diagrams is done and semantic similarity is computed and optimized using Sunflower optimization and Requirement traceability matrix is created. ...