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	ICSE	Conference
S2	An Empirical Study on Source Code Feature Extraction in Preprocessing of IR-Based Requirements Traceability	Bangchao Wang Yang Deng Ruiqi Luo Huan Jin	2022	QRS	Conference
S3	Configuring Latent Semantic Indexing for Requirements Tracing	Sebastian Eder Henning Femmer Benedikt Hauptmann Maximilian Junker	2015	ICSE	Conference
S4	IRRT: An Automated Software Requirements Traceability Tool based on Information Retrieval Model	Sen Zhang Hongyan Wan Yong Xiao Ziruo Li	2022	QRS	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	A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System	Khalid Mahmood Hironao Takahashi Mazen Alobaidi	2015	ISADS	Conference
S12	An IR-based Artificial Bee Colony Approach for	Danissa V. Rodriguez	2020	ICTAI	Conference

	Traceability Link Recovery	Doris L. Carver			
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	Journal
S15	Combining VSM and BTM to improve requirements trace links generation	Bangchao Wang Rong Peng Zhuo Wang Yaxin Zhao	2019	SEKE	Conference
S16	TRIAD: Automated Traceability Recovery based on Biterm-enhanced Deduction of Transitive Links among Artifacts	Hui Gao Hongyu Kuang Wesley K. G. Assunção Christoph Mayr-Dorn Guoping Rong He Zhang Xiaoxing Ma Alexander Egyed	2023	ICSE	Conference
S17	Visualizing Software Repositories through Requirements Trace Links	Kadir Ersoy Ecenur Sezer Susan Üsküdarlı Fatma Ba sak Aydemir	2023	RE	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	ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques	Emma Effa Bella Stephen Creff Marie-Pierre Gervais Reda Bendraou	2019	EDOC	Conference
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 to Code Traceability Creation by Code Comments	Guohua Shen Haijuan Wang Zhiqiu Huang YaoShen Yu Kai Chen	2021	IJSEKE	Journal
S27	SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents	Ritika Jain Smita Ghaisas	2014	ICSE	Conference

		Ashish Sureka			
S28	Evolving Software Trace Links between Requirements and Source Code	Mona Rahimi Jane Cleland-Huang	2018	ICSE	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	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	ICSE	Conference
S35	On the role of semantics in automated requirements tracing	Anas Mahmoud Nan Niu	2015	RE	Journal

1.2. Extracted Data for RQ2

Index	Title	IR Model	Stage	Scenarios
S1	An empirical study on recovering requirement-to-code links	VSM	Preprocessing Stage	Generation
S2	An Empirical Study on Source Code Feature Extraction in Preprocessing of IR-Based Requirements Traceability	VSM LSI	Preprocessing Stage	Generation
S3	Configuring Latent Semantic Indexing for Requirements Tracing	LSI	Links Generation Stage	Maintenance
S4	IRRT: An Automated Software Requirements Traceability Tool based on Information Retrieval Model	VSM	Links Generation Stage Links Refinement Stage	Generation
S5	Analyzing closeness of code dependencies for improving IR-based Traceability Recovery	VSM LSI JS	Links Refinement Stage	Generation
S6	An empirical study on the importance of source code entities for requirements traceability	LSI TM(LDA)	Preprocessing Stage	Validation
S7	Analyzing close relations between target artifacts for improving IR-based requirement traceability recovery	VSM	Links Generation Stage	Generation
S8	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description	LSI	Preprocessing Stage	Generation
S9	Achieving better requirements to code traceability: which refactoring should be done first?	VSM LSI	Preprocessing Stage	Generation
S10	Using Consensual Biterms from Text Structures of Requirements and Code to Improve IR-Based Traceability Recovery	VSM LSI JS	Links Refinement Stage Preprocessing Stage	Generation
S11	A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System	IR-based+	Links Generation Stage	Generation

S12	An IR-based Artificial Bee Colony Approach for Traceability Link Recovery	IR-based+	Links Generation Stage	Generation
S13	Propagating frugal user feedback through closeness of code dependencies to improve IR‐based traceability recovery	VSM LSI JS	Links Refinement Stage Preprocessing Stage	Validation
S14	Leveraging BPMN particularities to improve traceability links recovery among requirements and BPMN models	LSI	Links Generation Stage	Generation
S15	Combining VSM and BTM to improve requirements trace links generation	VSM TM(BTM)	Links Generation Stage	Generation
S16	TRIAD: Automated Traceability Recovery based on Biterm-enhanced Deduction of Transitive Links among Artifacts	VSM LSI JS	Links Generation Stage	Generation
S17	Visualizing Software Repositories through Requirements Trace Links	IR-based+	Links Generation Stage	Application
S18	Multi-Objective Information Retrieval-Based NSGA-II Optimization for Requirements Traceability Recovery	IR-based+	Links Generation Stage	Generation
S19	Filtering of false positives from IR-based traceability links among software artifacts	VSM	Links Refinement Stage	Validation
S20	Quality improvements for trace links between source code and requirements	IR-based+	Links Generation Stage	Generation
S21	Evaluation of Natural Language Processing for Requirements Traceability	VSM	None	Validation
S22	ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques	VSM LSI TM(LDA)	Links Generation Stage	Validation
S23	An Automated Hybrid Approach for Generating Requirements Trace Links	VSM TM(BTM)	Links Generation Stage	Generation
S24	Evaluating the Effectiveness of Various IR Models for Requirements Traceability	VSM LSI JS	None	Generation
S25	Search-Based Requirements Traceability Recovery： A Multi-Objective Approach	IR-based+	Links Generation Stage	Maintenance
S26	Supporting Requirements to Code Traceability Creation by Code Comments	VSM	Preprocessing Stage	Maintenance
S27	SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents	TM(LDA)	Links Generation Stage	Generation
S28	Evolving Software Trace Links between Requirements and Source Code	VSM LSI	Links Generation Stage	Maintenance
S29	Interactive recovery of requirements traceability links using user feedback and configuration management logs	VSM	Links Generation Stage Links Refinement Stage	Maintenance
S30	Supporting requirements to code traceability through refactoring	VSM LSI	Preprocessing Stage	Maintenance
S31	Recovering traceability links between requirements and source code using the configuration management log	VSM	Preprocessing Stage Links Generation Stage Links Refinement Stage	Maintenance
S32	Exploiting Parts-of-Speech for Effective Automated Requirements Traceability	VSM JS	Links Refinement Stage	Generation
S33	Using Frugal User Feedback with Closeness Analysis on Code to Improve IR-Based Traceability Recovery	VSM LSI JS	Links Refinement Stage	Generation
S34	Adaptive User Feedback for IR-Based Traceability Recovery	VSM	Links Refinement Stage	Validation
S35	On the role of semantics in automated requirements tracing	VSM	Links Generation Stage	Generation

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 Empirical Study on Source Code Feature Extraction in Preprocessing of IR-Based Requirements Traceability	Code Feature Extraction, Annotation Importance Assessment, and Annotation Redundancy Removal
S3	Configuring Latent Semantic Indexing for Requirements Tracing	Heuristic Measures
S4	IRRT: An Automated Software Requirements Traceability Tool based on Information Retrieval Model	Code Class Structure

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	A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System	Integrating semantic similarity
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	TRIAD: Automated Traceability Recovery based on Biterm-enhanced Deduction of Transitive Links among Artifacts	Consensual Biterms and Transitive Relationships
S17	Visualizing Software Repositories through Requirements Trace Links	None
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	ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques	Semi-Supervised Techniques
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 to Code Traceability Creation by Code Comments	Code Comments
S27	SANAYOJAN A framework for traceability link recovery between use-cases in software requirement specification and regulatory documents	None
S28	Evolving Software Trace Links between Requirements and Source Code	Trace Link Evolver
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	On the role of semantics in automated requirements tracing	Semantic Augmentatio

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 Empirical Study on Source Code Feature Extraction in Preprocessing of IR-Based Requirements Traceability	Requirements Use cases Use cases	Code Code Code	iTrust eTOUR Albergate

				EasyClinic SMOS
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	IRRT: An Automated Software Requirements Traceability Tool based on Information Retrieval Model	Requirements	Code	iTrust
S5	Analyzing closeness of code dependencies for improving IR-based Traceability Recovery	Requirements	Code	iTrust Gantt 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	A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System	Requirements	Requirements	Borland CaliberRM
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	TRIAD: Automated Traceability Recovery based on Biterm-enhanced Deduction of Transitive Links among Artifacts	Requirements Use cases Requirements	Code Code Requirements	Dronology WARC EasyClinic

				EBT Libest
S17	Visualizing Software Repositories through Requirements Trace Links	Requirements Requirements Requirements	Issues Requests Commits	public GitHub repository of a group of computer engineering students for their software engineering course
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 Code	Mylyn iTrust
S21	Evaluation of Natural Language Processing for Requirements Traceability	Requirements	Requirements	National Aeronautics and Space Administration (NASA)
S22	ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques	Requirements	Models	Aggreg0 Aggreg1 Aggreg2 Aggreg3
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 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 to Code Traceability Creation by Code Comments	Use Cases	Code	eTour iTrust
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	Evolving Software Trace Links between Requirements and Source Code	Requirements	Code	Domain Analysis App DOTS File Generator Apache Cassandra Database System Dronology
S29	Interactive recovery of requirements traceability links using user feedback and configuration management logs	Requirements	Code	An enterprise system
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 Requirements	Code Code JSP Code Requirements	Easy-Clinic i-Trust Modis
S35	On the role of semantics in automated requirements tracing	Requirements	Code Design	iTrust eTour CM-1

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 Empirical Study on Source Code Feature Extraction in Preprocessing of IR-Based Requirements Traceability	Threshold	Recall Precision F-Measure	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	IRRT: An Automated Software Requirements Traceability Tool based on Information Retrieval Model	Threshold	Recall Precision F-Measure	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
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	A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System	Threshold	Recall Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).
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	TRIAD: Automated Traceability Recovery based on Biterm-enhanced Deduction of Transitive Links	Threshold	Recall Precision	Level 3. Evidence obtained from academic studies (e.g., controlled lab experiments).

	among Artifacts		F-Measure AP MAP	
S17	Visualizing Software Repositories through Requirements Trace Links	Threshold	Recall Precision F-Measure	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	ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques	Threshold	Recall Precision F-Measure	Level 4. Evidence obtained from industrial studies (e.g., causal case studies in an industrial setting).
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 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).
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	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).
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	On the role of semantics in automated requirements tracing	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

Category	Strategy	IR model						Applying Phrase	Scenarios	Strategy Characteristics
		VSM	LSI	JS	LDA	BTM	IR-based+			
Artifact Text	Verb-object Phrases [S1]	●						P	G	Extracting verb-object phrases as main information and essential meaning.
	Improved Term Weighting Scheme [S6]		●		●			P	V	Proposing an improved term weighting scheme, namely, Developers Preferred Term Frequency/Inverse Document Frequency (DPTF/IDF) .
	Refactoring [S9][S30]	●	●					P	G,M	Solving the problem of missing symbols, misplaced symbols and repeated symbols.
	Annotation extraction [S2]	●	●					P	G	Introducing different types of comments to some extent compensate for vocabulary mismatches between requirements and source code to improve the accuracy of tracing links.
	Consensual Biterms [S10]	●	●	●				P	G	Extracting consensual biterms to first enrich the corpus for IR techniques.
	Code Comments [S26]	●						P	M	Introducing different types of comments to some extent compensate for vocabulary mismatches between requirements and source code to improve the accuracy of tracing links.
	Semantic Augmentation [S35]	●						R	G	By utilizing the Vector Space Model (VSM) with additional support from domain-specific thesauri and general-purpose thesauri such as WordNet.
	Trace Link Evolver (TLE) [S28]	●	●					R	M	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.
	Configuration Management Log [S29][S31]	●						R	M	Restoring links by finding revisions in the configuration management log that contain words related to requirements.
	Integrating semantic similarity [S11]						●	R	G	This strategy leverages external knowledge bases, such as DBpedia and BabelNet, to enrich the textual information of requirements artifacts.
	Global and Local Weight [S10]	●	●	●				F	G	Using consensual biterms to adjust global and local weight to adjust the ranking of candidate lists.
Artifact Structural	Traceability Rules [S8]		●					P	G	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
	Commonality and Variability Analysis (CVA) [S31]	●						P	M	Analyzing to which products elements (e.g., requirements, code elements) belong.
	Analyzing Closeness of Code Dependencies [S5] [S13]	●	●	●				P,F	G,V	Quantifying the interaction degree of call dependency and data dependency between two code classes.
	Biterm Extraction Strategies [S16]	●	●	●				R	G	Extracting pairs of terms (biterms) from documents, which are then used to capture co-occurring concepts within short texts.
	Analyzing Close Relations [S7]	●						R	G	Calculating the close relations (semantic similarity) between target artifacts
	Code class Structure [S4]	●						F	G	Utilizing the hierarchical and relational structure of code classes, such as inheritance or associations between classes.
	Correlation Among Classes [S19]	●						F	V	Using structural or co-changing dependencies or both to find correlations between classes and use these dependencies to verify traceability links.
Model-based	Artificial Bee Colony Algorithm [S12]						●	R	G	An optimization technique inspired by honey bee foraging behavior, used to efficiently explore and improve potential solutions in trace link recovery.
	Genetic Algorithm [S18][S20][S23][S25]	●				●	●	R	G,M	A search heuristic inspired by the process of natural selection.
	Heuristic Measures [S3]		●					R	M	A fully automated technique to determine appropriate configurations for LSI to recover links between requirements artifacts.
	Hybrid Method [S15][S23]	●				●		R	G	Combing VSM and BTM which can help relieve data sparsity caused by short text.
	Semi-Supervised Techniques [S22]	●	●		●			R	V	Utilizing a combination of semi-supervised learning techniques and NLP to enhance IR-based traceability recovery.
	Classification [S31]	●						F	M	Classifying traceability links into 5 five types using the CVA results, then using the classification to refine links.
Others	BPMN-specific approaches [S14]		●					R	G	An approach that improves the results of TLR between requirements and BPMN models.
	Adaptive User Feedback [S34]	●						F	V	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.
	ConPOS Approach [S32]	●		●				F	G	Pruning trace links using the primary POS classification and apply constraints to recovery as a filtering process.
	User Feedback [S29]	●						F	M	Introducing user validation for candidate links to improve accuracy
	Frugal User Feedback with Closeness Analysis on Code [S13][S33]	●	●	●				F	V,G	Introducing only a small amount of user feedback into the closeness analysis on call and data dependencies in code.

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	16	http://www.coest.org/	[S2] [S4] [S5] [S6] [S7] [S9] [S10] [S13] [S19] [S20] [S26] [S30] [S32] [S33] [S34] [S35]
	Requirements (50)	Code (299)	14950	314				
	Use cases (33)	JSP (47)	1551	58	Small			
eTour	Use cases (58)	Code (116)	6728	308	Large	10	http://www.coest.org/	[S1] [S2] [S9] [S12] [S18] [S23] [S25] [S26] [S30] [S35]
	Requirements (58)	Code (116)	6728	366				
EasyClinic	Requirements (30)	Code (47)	1410	83	Small	8	http://www.coest.org/	[S1] [S2] [S3] [S7] [S15] [S16] [S23][S34]
	Use cases (30)	Test cases (63)	1890	63				
	UML interaction diagram (20)	Code classes (47)	940	69				
Gantt	Requirements (17)	Code (55)	935	54	Small	6	http://www.ganttproject.biz	[S5] [S6] [S10] [S13] [S33]
	Requirements (16)	Code (124)	1984	315			https://github.com/bardsoftware/ganttproject	
	Use cases (67)	Code (100)	6700	1044			http://www.coest.org/	[S7]
EBT	Requirements (40)	Code (50)	2000	98	Small	5	http://www.coest.org/	[S12] [S15] [S16] [S18] [S23]
	Requirements (16)	Code (124)	1173	315				
Pig	Requirements (Unclear)	Code (Unclear)	Unclear	Unclear	Unclear	4	https://pig.apache.org/	[S10] [S13] [S24] [S34]
	Requirements (87)	Code (289)	25143	547	Large		https://github.com/apache/pig	
	Requirements (58)	Code (754)	43732	Unclear				
Albergate	Requirements (82)	Code (1771)	145222	871	Small	4	http://www.coest.org/	[S2] [S12] [S18] [S25]
WARC	Functional requirements (43)	Software requirements specification (89)	3827	78	Large	4	http://www.coest.org/	[S15] [S16] [S23] [S23]
	High-level Requirements (17)	Low-level Requirements (69)	1173	68	Small			
Derby	Requirements (390)	Code (611)	238290	2315	Large	3	https://github.com/apache/derby	[S10] [S13] [S24]
	Requirements (133)	Code (2184)	290472	Unclear	Large			
Infinispan	Requirements (116)	Code (413)	47908	744	Large	3	http://infinispan.org/	[S10] [S13] [S33]
	Requirements (232)	Code (319)	74008	1116	Large		https://github.com/infinispan/infinispan	
CM-1	High-level Requirements (235)	Design (220)	51700	361	Large	3	http://www.coest.org/	[S3] [S7] [S35]
	Requirement (298)	Code (90)	26820	546				
Maven	Requirements (68)	Code (236)	16048	356	Large	3	http://maven.apache.org/	[S10] [S13] [S33]
	Requirements (36)	Code (82)	2880	151	Small		https://github.com/apache/maven	
MR	Use cases (60)	Test cases (24)	1,440	711	Small	3	Unclear	[S3]
	Use cases (28)	Defect reports (135)	3,780	1,422	Small			[S3]
	Use cases (21)	Change requests (28)	588	396	Small			[S3]
Drools	Requirements (183)	Code (248)	45,384	841	Large	2	https://github.com/kiegroup/drools	[S10] [S13]
Groovy	Requirements (104)	Code (100)	10,400	180	Large	2	https://github.com/apache/groovy	[S10] [S13]
MODIS	Requirements (26)	Code (521)	13,546	229	Large	2	Unclear	[S3] [S34]
Seam	Requirements (189)	Code (150)	28350	463	Large	2	http://www.seamframework.org/Seam2.html	[S10] [S13]
Pooka	Requirements (90)	Code (298)	26,820	507	Large	2	http://www.suberic.net/pooka/	[S6] [S32]
Dronology	Requirements (58)	Code (184)	8,584	Unclear	Large	2	https://dronology.info/datasets/	[S16] [S28]
Cassandra	Requests (65)	Code (328)	21,320	Unclear	Large	2	http://tinyurl.com/TLEArtifacts	[S24] [S28]

Note: There are 32 open-source datasets with one frequency, i.e., SMOS, Lucene, Mylyn, Lynx, jHotDraw (JHD), SMS, Pine, LEDA, Network Control System, Domain Analysis App, Car Rental System, Customer Relationships System, Hive, Mina, Solr, Synapse, Tika, ActiveMQ, SIP Communicator, CUnit, Libest, Borland CaliberRM, aggreg0-3, ibooks, Industrial case study , Academic case study, Software Engineering Course, IT services company, DOTS File Generator, an enterprise system.

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 Xplore	323	128	195	574	1925
Engineering Village	1604	692	912		
Springer	118	36	82		
Science Direct	205	47	158		
ACM	91	8	83		
Google scholar	1640	571	1069		

2.1. Search records

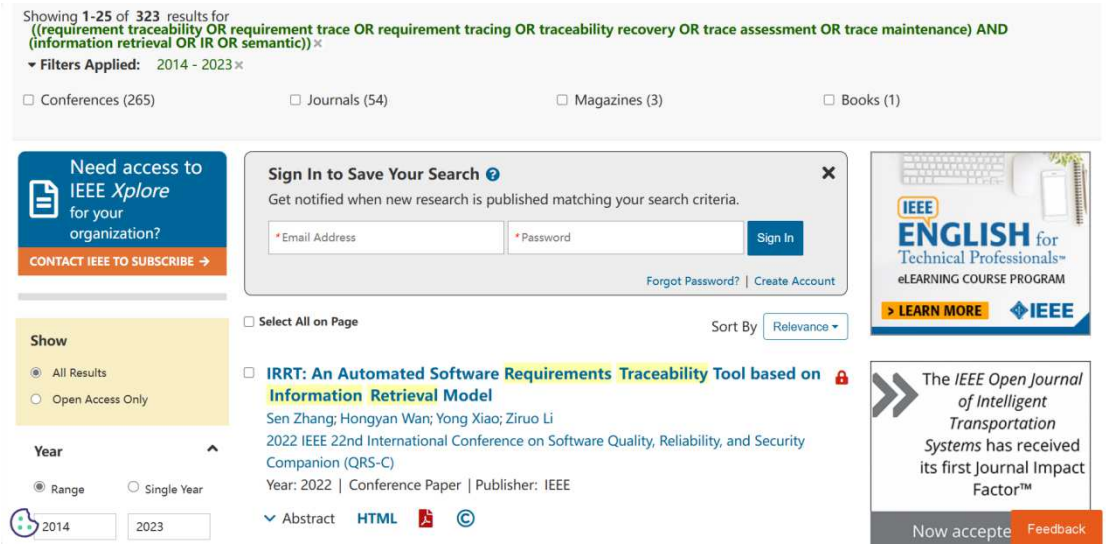
Search terms:

P1	requirement traceability
P2	requirement trace
P3	requirement tracing
P4	traceability recovery
P5	trace assessment
P6	trace maintenance
I1	information retrieval
I2	IR
I3	semantic

(1) IEEE Xplore

	Command Search
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	323
Total	323

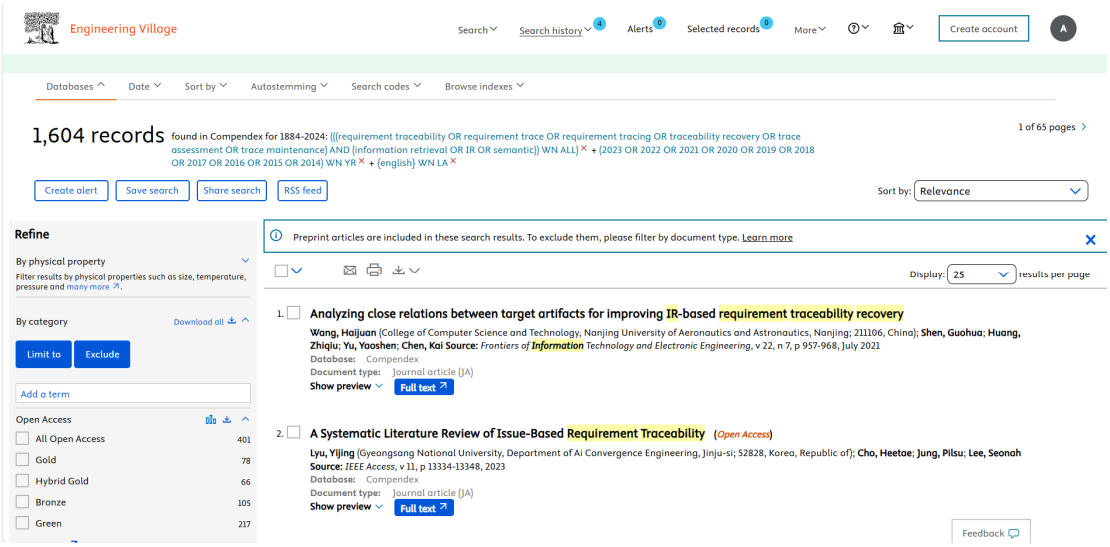
The screenshot of search process in IEEE Xplore：



(2) Engineering Village

	Abstract + Title + Keywords (Index term)
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	1604

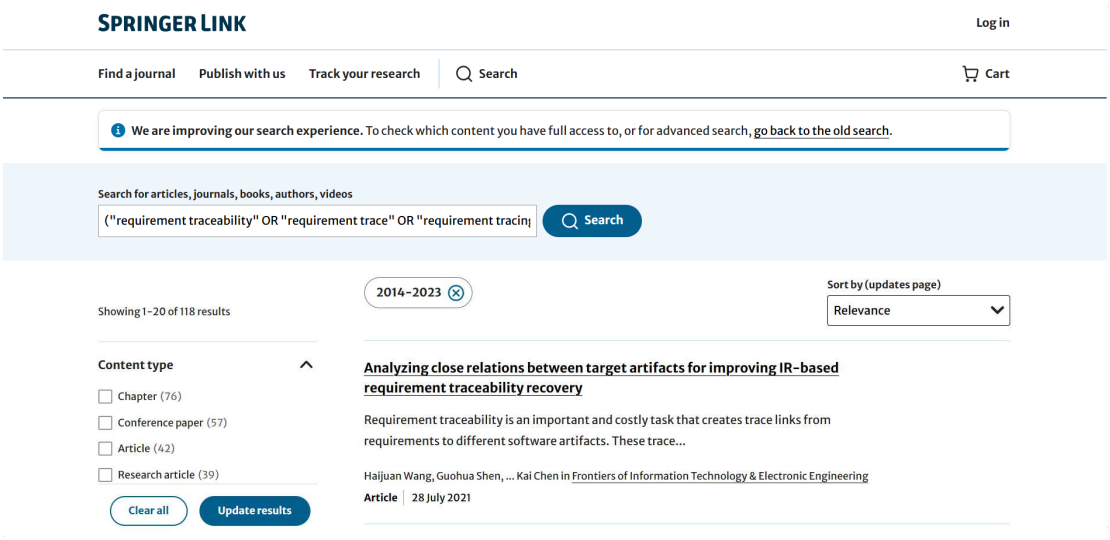
The screenshots of search process in EI



(3) Springer

	Abstract + Title + Keywords (Index term)
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	118

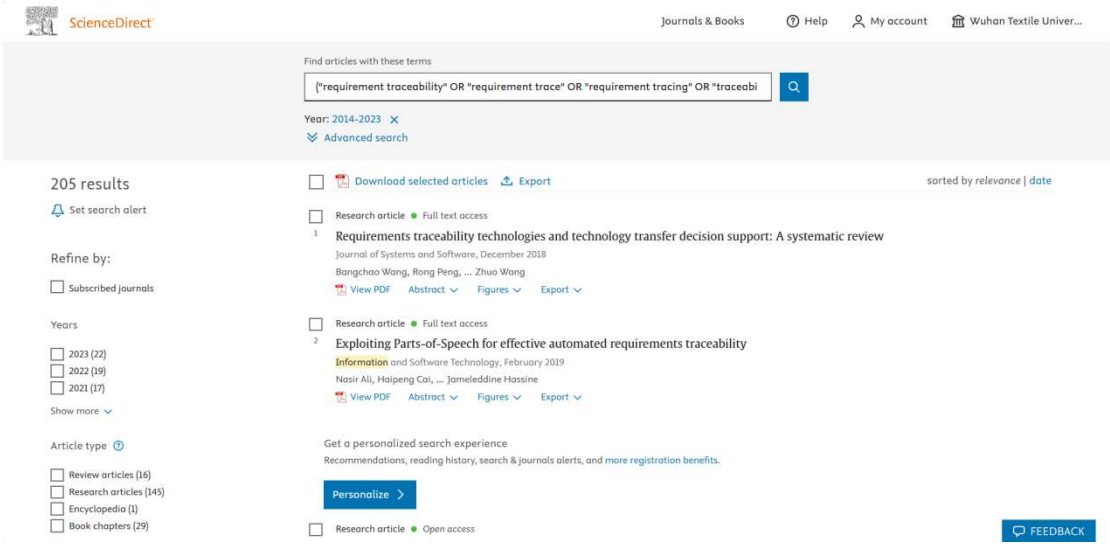
The screenshots of search process in Springer:



(4) Science Direct

	Abstract + Title + Keywords (Index term)
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	205

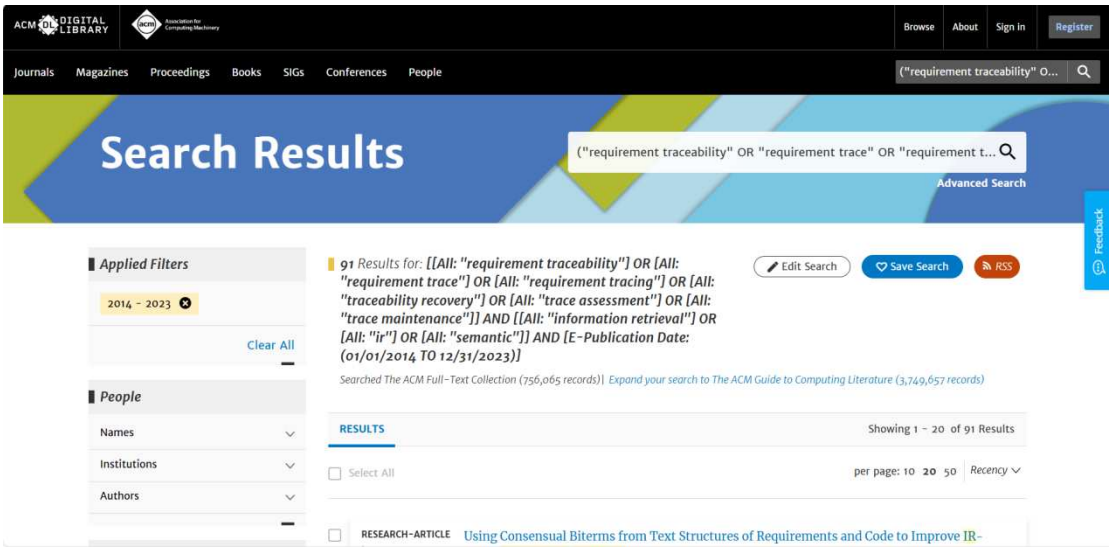
The screenshots of search process in Science Direct



(5) ACM Digital Library

	Anywhere
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	91

An example screenshots of search process in ACM Digital Library:



(6) Google Scholar

	Anywhere
(P1 OR P2 OR P3 OR P4 OR P5 OR P6) AND (I1 OR I2 OR I3)	1640
Total	1640

An example screenshots of search process in Google Scholar