Describing and Accessing Software Engineering Research Knowledge

zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Angelika Kaplan

aus Karlsruhe

Tag der mündlichen Prüfung: 22. Dezember 2025
Erster Gutachter: Prof. Dr. Ralf H. Reussner
Zweiter Gutachter: Prof. Dr. Michael Goedicke

Abstract

Software Engineering (SE) research drives progress and innovation in science, practice, and society, forming a cornerstone of the digital world. Simultaneously, the digital transformation of the scientific ecosystem is reshaping how research knowledge is acquired, shared, and reused, promoting the principles of Open Science and FAIR (Findable, Accessible, Interoperable, and Reusable).

Despite these advancements, scientific papers, typically disseminated as `PDF` documents, remain the primary medium for communicating research findings. These documents are inherently monolithic and lack machine-readable semantics, which hinders automated processing and effective research knowledge retrieval. In case of SE research, such publications are often further accompanied by supplementary materials, including data and software, which provide the empirical foundation and evidence for validating research claims. These materials are often (if available) linked or stored in a heterogeneous manner, further complicating research data management and the realization of the FAIR principles.

This doctoral thesis addresses these weaknesses by developing an approach that semantically describes SE research artifacts and enables access to research knowledge. The approach's architecture, using Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), comprises two core modules: (1) a *knowledge description module* that supports structured schema representation through semi-automated population of knowledge graphs, and (2) a *knowledge access module* that enables schema-agnostic retrieval.

Central to our approach is a conceptual classification and data schema designed to describe research findings in scientific publications by their validity and supporting evidence. We instantiated this schema in the research field of software architecture and evaluate its feasibility through a literature study of top-tier publications. Furthermore, we propose a general research process based on the GQM approach (Goal, Question, Metric) that integrates descriptive and normative aspects of research artifacts and their knowledge statements, and, thus, supports schema evolution and long-term sustainability.

To operationalize our conceptual approaches, we use the FAIR-compliant Open Research Knowledge Graph (ORKG), deriving ORKG templates as blueprints for schema specification and for graph population. We develop a flexible classification framework that supports transfer learning and prompting to train classifiers for semi-automated knowledge graph population based on large language models. The framework is also able to address challenges such as dataset imbalance via data augmentation and text perturbation techniques. The feasibility and maturity of our framework are validated using the labeled dataset of the aforementioned literature study as gold standard dataset for benchmarking. This dataset also serves as the basis to populate the knowledge graph. Besides tracking and assessing conventional performance metrics for the automated classification task, we defined combined sustainability metrics within our framework to enable trade-off decisions between performance and carbon emissions.

To facilitate access to the enriched research knowledge, we implement a schema-agnostic, training-free retrieval approach with provenance tracking that is capable of adapting to different and evolving schemas. This approach is evaluated against three state-of-the-art baselines from top-tier ACL (Association for Computational Linguistics) venues, using a custom question-answering dataset constructed based on our retrieval question taxonomy. The evaluation considers question complexity, realistic use cases, and differently typed questions as well as different knowledge base variants to assess the robustness of our approach. The results show superior performance across most traditional retrieval and LLM-as-a-Judge metrics compared to the baselines, especially for complex user questions, while indicating the need for further improvements regarding answer generation quality.

In this doctoral thesis, we reflect on and analyze current research practices, contribute to meta-research aimed at calibrating the scientific ecosystem and supporting the sustainable management of research artifacts in SE. In conclusion, our (maintainable) schema and classification framework facilitate the construction of a machine-processable knowledge base, which serves as the foundation for a question-answering system. By integrating the complementary strengths of knowledge graphs and large language models, the approach advances both the state-of-practice and the state-of-the-art in semantic research knowledge description and access.

Zusammenfassung

Die Forschung im Bereich Software Engineering (SE) treibt Fortschritt und Innovationen in Wissenschaft, Industrie und Gesellschaft voran und bildet ein zentrales Fundament des digitalen Zeitalters. Gleichzeitig verändert die digitale Transformation die Forschungslandschaft, insbesondere hinsichtlich der Art und Weise, wie Forschung durchgeführt wird, Ergebnisse und Erkenntnisse geteilt und wiederverwendet werden. Diese Entwicklungen fördern die Prinzipien und Richtlinien von Open Science und FAIR (engl. Findable, Accessible, Interoperable, Reusable), die auf eine nachhaltige Forschungspraxis abzielen.

Trotz dieser Fortschritte bleiben wissenschaftliche Publikationen altbewährt, meist in Form von PDF-Dokumenten, das primäre Kommunikationsmedium von Forschungsergebnissen. Diese Formate sind jedoch monolithisch aufgebaut und nicht maschinenlesbar, was die automatisierte Verarbeitung, Integration und den gezielten Zugriff auf Forschungswissen erheblich erschwert. In der SE-Forschung werden solche Publikationen häufig durch ergänzende Artefakte wie Datensätze und Software ergänzt, die zur empirischen Validierung von Forschungsergebnissen dienen. Diese Artefakte sind jedoch oft unzureichend verlinkt oder uneinheitlich gespeichert, was die Umsetzung eines nachhaltigen Forschungsdatenmanagements sowie der FAIR-Prinzipien erschwert.

Diese Dissertation adressiert die genannten Schwächen durch die Entwicklung eines konzeptionellen und technischen Ansatzes zur semantischen Beschreibung und zum schemaunabhängigen Zugriff auf SE-Forschungsartefakte (Publikationsinhalte, Daten und Software). Der entwickelte Ansatz basiert auf Retrieval-Augmented Generation (RAG) und umfasst zwei zentrale Module: (1) ein *Wissensbeschreibungsmodul* zur semi-automatisierten Befüllung von Wissensgraphen und (2) ein *Wissenszugangsmodul* mit schemaunabhängigen Retrieval-Funktionalitäten.

Im Zentrum des Ansatzes steht ein Klassifikations- und Datenschema zur strukturierten Beschreibung von SE-Publikationen auf Basis ihrer Forschungsergebnisse, deren Validität und der zugrunde liegenden Evidenz. Dieses Schema wurde exemplarisch für das Forschungsfeld Softwarearchitektur instanziiert und im Rahmen einer systematischen Literaturstudie mit Veröffentlichungen führender Fachkonferenzen evaluiert. Dabei wurde ein Goldstandard-Datensatz erstellt, der sowohl zur Evaluation als auch zur Befüllung des Wissensgraphen dient. Zur Unterstützung der Weiterentwicklung des Schemas wurde ein allgemeiner Forschungsprozess auf Basis des GQM-Paradigmas (engl. Goal, Question, Metric) entwickelt, der sowohl deskriptive als auch normative Aspekte integrieren kann.

Zur technischen Umsetzung der konzeptionellen Ansätze wird der FAIR-konforme Open Research Knowledge Graph (ORKG) genutzt. Aus dem Klassifikationsschema wurden ORKG-Templates abgeleitet, die als wiederverwendbare Vorlagen für die strukturierte Repräsentation von Forschungsartefakten dienen. Ergänzend wurde ein flexibles Klassifikationsframework entwickelt, das Transfer Learning und Prompting unterstützt, um robuste Klassifikatoren basierend auf großen Sprachmodellen (engl. Large Language Models, LLMs)

zu trainieren. Dieses Framework adressiert typische Herausforderungen wie unausgewogene Datensätze und ermöglicht eine kontinuierliche Anpassung an sich weiterentwickelnde Schemata. Zur Bewertung der Klassifikationsleistung wurden neben klassischen Metriken auch kombinierte Nachhaltigkeitsmetriken eingeführt, um Trade-off-Entscheidungen zwischen Modellleistung und CO_2-Emissionen zu ermöglichen.

Für den Zugriff auf das angereicherte und semantisch strukturierte Forschungswissen wurde ein schemaunabhängiger, trainingsfreier Retrieval-Ansatz mit Provenienzverfolgung implementiert. Dieser Ansatz passt sich flexibel an unterschiedliche und sich entwickelnde Schemata an und wurde gegen drei State-of-the-Art-Baselines aus führenden ACL-Veröffentlichungen (Association for Computational Linguistics) evaluiert. Die Evaluation basiert auf einem eigens entwickelten Frage-Antwort-Datensatz und berücksichtigt realistische Anwendungsszenarien, unterschiedliche Fragetypen sowie Varianten der Wissensbasis zur Bewertung der Robustheit. Die Evaluationsergebnisse zeigen, dass der entwickelte Ansatz in den meisten klassischen Retrieval-Metriken sowie in LLM-as-a-Judge-Bewertungen eine überlegene Leistung gegenüber den Baselines erzielt, insbesondere bei komplexen Retrieval-Operationen. Gleichzeitig identifizieren wir Potenziale zur Verbesserung der Antwortgenerierung, die in zukünftiger Arbeit adressiert werden sollen.

In dieser Dissertation reflektieren wir aktuelle Forschungspraktiken und leisten einen Beitrag zur Meta-Forschung mit dem Ziel, das Forschungsfeld zu verbessern und das nachhaltige Management von Forschungsartefakten im SE-Bereich zu fördern. Unser wartbares Klassifikationsschema bildet die Grundlage für den Aufbau einer maschinenverarbeitbaren Wissensbasis sowie eines Frage-Antwort-Systems. Durch die Verbindung der komplementären Stärken von Wissensgraphen und großen Sprachmodellen trägt der Ansatz sowohl zur Weiterentwicklung des Forschungsstandes als auch zur Verbesserung etablierter Forschungspraktiken bei.

Contents

Abstract i

Zusammenfassung iii

List of Figures xi

List of Tables xiii

I. Preliminaries 1

1. Introduction **3**
1.1. Background and Problem Statements 3
1.2. Research Goal and Main Research Question 7
1.3. Envisioned Solution and Contributions 9
1.4. Research Overview 11
1.5. Roles and Perspectives 14
1.6. Thesis Outline 15

2. Foundations **17**
2.1. Terminology 17
2.1.1. Scientific Ecosystem and Meta-Research 17
2.1.2. Software Engineering Research Terminology 19
2.1.3. Open Science and FAIR Principles 21
2.2. From Artificial Intelligence to Generative AI 22
2.3. Natural Language Processing and Understanding 23
2.3.1. Large Language Models 24
2.3.2. Retrieval-Augmented Generation 25
2.4. From Data to Knowledge and Wisdom 26
2.5. Knowledge Representation Approaches 27
2.5.1. Modeling Concepts 27
2.5.2. Modeling Techniques for Knowledge Representations 28
2.6. The Open Research Knowledge Graph 30
2.7. Sustainable Software Development and Green AI 33
2.8. Evaluation Preliminaries 34
2.8.1. Goal-Question-Metric Approach 34
2.8.2. Properties for Language Evaluation 35
2.8.3. Inter-Annotator Agreement 36

2.8.4. Performance Metrics in Machine Learning . . . 38
2.8.5. Sustainability Aspects . . . 40
2.8.6. Metrics for Retrieval-Augmented Generation Systems . . . 41
2.8.7. The One-Factor-at-a-Time Method . . . 45
2.8.8. Competency Questions . . . 46

3. State of Practice . . . 47
3.1. Approaches for Assessing and Describing Software Engineering Research 47
3.1.1. ACM SIGSOFT Empirical Standards for Software Engineering . . 47
3.1.2. Review Quality Criteria . . . 48
3.1.3. Artifact Review and Badging Systems . . . 49
3.1.4. De Facto Standard Classifications in Software Engineering Research 50
3.1.5. Conclusion . . . 51
3.2. Approaches for Persisting and Accessing Software Engineering Research 52
3.2.1. The Evolution of (Web) Search . . . 52
3.2.2. Scientific Search Engines . . . 54
3.2.3. Version Control Systems and Data Repositories . . . 55
3.2.4. Conclusion . . . 56
3.3. Vision of FAIR and Sustainable Research Data Management . . . 57

II. Knowledge Description in Software Engineering Research 59

4. Overview . . . 61

5. An Evaluation Approach for Classification and Data Schemas . . . 63
5.1. Background and Problem Statement . . . 64
5.2. Preliminaries for the Evaluation Approach . . . 65
5.2.1. Running Example . . . 65
5.2.2. Levels of Data and Knowledge Abstractions . . . 66
5.2.3. Types of Classification Problems . . . 67
5.3. Evaluation Process and Metrics . . . 69
5.3.1. Step 1: Evaluating the Structure's Suitability . . . 70
5.3.2. Step 2: Evaluating the Applicability . . . 74
5.3.3. Step 3: Evaluating the Purpose . . . 77
5.3.4. Summary and Limitations . . . 81
5.4. Application of the Evaluation Approach . . . 82
5.5. Threats to Validity . . . 84
5.6. Conclusion . . . 85

6. A Sustainable Documentation Process for Literature Studies . . . 87
6.1. Background and Problem Statement . . . 88
6.2. Sustainable Literature Studies . . . 89
6.3. Tool Support and Machine-readable Formats for Literature Studies . . . 90
6.3.1. The SLR Toolkit . . . 90

6.3.2. Version Control Systems . . . 91
6.4. Our Proposal for a Tool-supported Documentation Process . . . 92
6.4.1. Our Approach from the Perspective of Leading Authors . . . 93
6.4.2. Our Approach from the Perspective of Data Extractors . . . 96
6.5. Feature Analysis and Evaluation . . . 98
6.5.1. Feature Analysis Set for Tool Support . . . 99
6.5.2. Evaluation Results . . . 100
6.6. Conclusion . . . 103

7. A Classification and Data Schema for Software Engineering Research Artifacts . . . 105
7.1. Background and Problem Statement . . . 106
7.2. Previous Work . . . 106
7.3. Research Approach . . . 108
7.4. A Classification and Data Schema for Software Architecture Research . . . 109
7.5. Evaluation of the Schema . . . 110
7.5.1. Evaluation of the Structure's Suitability and Purpose . . . 111
7.5.2. Evaluation of the Applicability . . . 112
7.5.3. Summary and Findings . . . 112
7.5.4. Threats to Validity . . . 113
7.6. Review Protocol . . . 114
7.6.1. Data Sources and Search Strategy . . . 114
7.6.2. Study Selection . . . 115
7.6.3. Study Quality Assessment . . . 115
7.6.4. Data Extraction Form . . . 115
7.6.5. Data Extraction Process . . . 118
7.6.6. Data Synthesis . . . 118
7.6.7. Threats to Validity . . . 119
7.7. Results of the Literature Review . . . 120
7.8. Findings and Discussion . . . 120
7.9. Proposals for the Software Architecture Research Community . . . 127
7.10. Limitations . . . 130
7.11. Conclusion . . . 131

8. A General Research Process and Maintenance Approach . . . 133
8.1. Problem Statement . . . 134
8.2. Change Triggers for Classification Schemas in Research Contexts . . . 134
8.3. General Research Process and Maintenance Approach . . . 136
8.3.1. Overview of the General Research Process . . . 136
8.3.2. Adapted Evaluation Approach for Classification Schemas . . . 137
8.4. Research Approach and Review Protocol . . . 140
8.4.1. Research Aim and Questions . . . 140
8.4.2. Pilot Study . . . 140
8.4.3. Data Sources and Search Strategy . . . 141
8.4.4. Study Selection . . . 141
8.4.5. Study Quality Assessment . . . 142

8.4.6. Data Extraction Form and Documentation . . . 143
8.4.7. Threats to Validity . . . 143
8.5. Results of the Literature Review . . . 144
8.6. Findings and Discussion . . . 146
8.7. Conclusion . . . 149

III. Knowledge Access in Software Engineering Research 151

9. Overview . . . 153

10. KARAGEN: Knowledge Augmented Retrieval and Generation Engine . . . 155
10.1. Approach and Problem Statement . . . 155
10.2. Quality Assurance for Experiments . . . 156
10.2.1. Experiment Management Tracking Tools . . . 157
10.2.2. Tracking Energy Consumption and Carbon Emissions . . . 158
10.2.3. Best Practices for Experiment Settings . . . 159
10.3. Question-Answering Taxonomy for Retrieval Systems . . . 161
10.4. Experiment Preliminaries . . . 163
10.4.1. Experimental Environment . . . 163
10.4.2. Gold Standard Dataset for Classification . . . 164
10.4.3. Gold Standard Question-Answering Dataset for Retrieval . . . 165
10.5. Conclusion . . . 166

11. Formalization of an ORKG Template for Software Architecture Research . . . 167
11.1. Problem Statement and Research Aim . . . 167
11.2. Semantic Structuring for Software Architecture Research . . . 168
11.3. Knowledge Graph Variants and Population . . . 170
11.4. Visulite: Visualization of Literature . . . 172
11.5. Conclusion . . . 173

12. A Classification Framework for Research Artifacts . . . 175
12.1. Problem Statement . . . 176
12.2. Challenges for an Automated Approach . . . 177
12.3. Classification Framework . . . 179
12.4. Evaluation . . . 182
12.4.1. Evaluation Settings . . . 183
12.4.2. Evaluation Results . . . 185
12.5. Threats to Validity . . . 197
12.6. Retrospective Analysis . . . 198
12.6.1. Background and Research Approach . . . 198
12.6.2. Structured Abstracts in Software Engineering . . . 199
12.6.3. Experience, Results, and Implications . . . 201
12.7. Conclusion . . . 202

13. A Retrieval Framework and Knowledge Access Module . . . **203**
13.1. Problem Statement . . . 203
13.2. HubLink: A Novel Retrieval Approach . . . 204
13.2.1. Indexing . . . 205
13.2.2. Retrieval . . . 206
13.2.3. Answer Generation and Information Linking . . . 208
13.3. Retrieval Framework . . . 208
13.4. Evaluation . . . 209
13.4.1. Evaluation Settings . . . 210
13.4.2. Evaluation Results . . . 212
13.5. Threats to Validity . . . 220
13.6. Conclusion . . . 221

IV. Reflections **223**

14. Related Work . . . **225**
14.1. Knowledge Description and Representation . . . 225
14.1.1. Evaluation Approaches and Metrics for Classification Schemas . 225
14.1.2. Guided Tool Support for Literature Studies in Software Engineering 230
14.1.3. Classification Schemas and Meta-Research Studies in Software Architecture Research . . . 231
14.1.4. Unified Approaches in Software Engineering Research . . . 236
14.2. Knowledge Persistence and Access . . . 239
14.2.1. Methodological Preliminaries . . . 239
14.2.2. Open Research Knowledge Graph Template Specifications . . . 241
14.2.3. Automated Classification Approaches for Software Engineering Research Artifacts . . . 244
14.2.4. Retrieval Approaches . . . 248

15. Conclusion . . . **253**
15.1. Summary . . . 253
15.2. Relevance and Implications for Research and Practice . . . 258
15.3. Outlook . . . 261

Bibliography . . . **267**

A. Appendix . . . **307**
A.1. Contribution C3: Artifact Availability of Case Studies . . . 307
A.2. Contribution C5: ORKG Template Variants . . . 308
A.3. Contribution C6: Additional Results for Classification . . . 309
A.3.1. Evaluation Results for Hyperparameter Search . . . 309
A.3.2. Visualization of the Best Results for Hyperparameter Search . . . 310
A.3.3. Carbon Emission Results . . . 311

A.4. Contribution C7: Additional Results for Retrieval 313
A.4.1. Evaluation Results for Question Complexity Types 313
A.4.2. Evaluation Results for Use Cases 314
A.4.3. Evaluation Results for Semi-Typed and Untyped Questions 316
A.4.4. Evaluation Results for Hop Counts 317

List of Figures

1.1. General research life cycle . . . 4
1.2. KARAGEN: Knowledge Augmented Retrieval And Generation ENgine . . 9
1.3. Vision, goals, problems, research question, and contributions . . . 13

2.1. The translational framework for meta-research . . . 18
2.2. The basic Open Research Knowledge Graph schema . . . 32
2.3. The Open Research Knowledge Graph template system . . . 33
2.4. The Goal Question Metric (GQM) approach . . . 35
2.5. Relation between conceptualization, abstraction, modeling language, and model . . . 37

5.1. Taxonomies in software engineering research . . . 64
5.2. Running example . . . 66
5.3. Different levels of abstraction representing research artifacts . . . 67
5.4. Research process for evaluating classification and data schemas . . . 69

7.1. Structured knowledge representation for software engineering papers . . 107
7.2. Research approach for the software architecture classification schema . . 108
7.3. General classification schema for software engineering research . . . 109
7.4. Distribution of software architecture research objects . . . 121
7.5. Distribution of evaluation methods . . . 121
7.6. Relations between research objects, properties, evaluation methods, and replication artifacts . . . 123
7.7. Relations between evaluation methods, properties, threats to validity, and replication artifacts . . . 124
7.8. Overview of replication artifact availability . . . 126

8.1. General research process and maintenance approach . . . 136
8.2. Illustration of generality and appropriateness . . . 137
8.3. Illustration of generality w.r.t. the metrics laconicity and lucidity . . . 138
8.4. Illustration of appropriateness w.r.t. the metrics completeness and soundness 139
8.5. Feature diagram of general classification and data schema . . . 148

10.1. KARAGEN, highlighting the contributions C5–C7 . . . 156

11.1. An ORKG template for software architecture research . . . 171
11.2. Visulite: Visualization of Literature . . . 173

12.1. Overview of automated classification approaches . . . 178
12.2. Feature diagram of the classification framework . . . 180

12.3. DeBERTa hyperparameter search for *Property* with the augmented dataset 184

13.1. Overview of the retrieval approach HubLink 205

14.1. Development and evaluation process for taxonomies in Software Engineering 226
14.2. Taxonomy evaluation framework in Information Systems 227
14.3. Workflow for the development and generation of the SciQA benchmark . 240
14.4. Excerpt from the ORKG template for Requirements Engineering 242
14.5. Excerpt from the ORKG template for Engineering Science 243

A.1. Templates for different ORKG knowledge bases 308
A.2. XLNet hyperparameter search for *Research Object* with the augmented dataset 310
A.3. XLNet hyperparameter search for *Evaluation Method* with the augmented dataset 310
A.4. DeBERTa hyperparameter search for *Property* with the augmented dataset 310

List of Tables

2.1. Template statements for research objects and objectives 20
2.2. Overview of architecture types and the corresponding LLMs 24
2.3. Types of statements in the ORKG 31
2.4. Relation between the Kappa statistic and the strength of agreement . . . 38
2.5. Confusion matrix for binary classification 39

5.1. Orthogonality matrix of the mug taxonomy 73
5.2. System Usability Scale (SUS) 76
5.3. Citation analysis of our evaluation approach 83

6.1. Literature review process in software engineering research 89
6.2. Feature analysis list for literature study tools 100
6.3. Feature set scores and overall scores 103

7.1. Data extraction form 117
7.2. Overview of evaluation and threats to validity guidelines 127

8.1. Number of results by query and filtered for the database search strategy 142
8.2. Data extraction form to capture classifications for the description of research artifacts in software engineering research 144
8.3. Overview of included publications mapped to the main meta-research areas 147

12.1. Overview of hyperparameter search space 184
12.2. Evaluation results of trained models for the *Research Object* category using the basic dataset 186
12.3. Evaluation results of trained models for the *Evaluation Method* category using the basic dataset 187
12.4. Evaluation results of trained models for the *Property* category using the basic dataset 188
12.5. Evaluation results of trained models for the *Research Object* category using the augmented dataset 189
12.6. Evaluation results of trained models for the *Evaluation Method* category using the augmented dataset 190
12.7. Evaluation results of trained models for the *Property* category using the augmented dataset 191
12.8. Evaluation results of the zero-shot approach for different LLMs 193
12.9. Environmental impact of trained models in the category *Property* (combined metrics) 195

12.10. Environmental impact of the zero-shot approach in the category *Property* (combined metrics) 196

13.1. Evaluation results of the retrieval performance of HubLink variants and baseline approaches 214
13.2. Performance by question complexity 215
13.3. Evaluation results of the retrieval performance based on semi-typed and untyped questions 216
13.4. Evaluation results of the retrieval performance for different graph variants 217
13.5. Evaluation results of the answer generation quality 218

14.1. Quality assessment dimensions for knowledge graphs 228

A.1. Overview of case studies with available input data 307
A.2. Results of hyperparameter search 309
A.3. Carbon emissions for trained models (basic dataset) 311
A.4. Carbon emissions for trained models (augmented dataset) 311
A.5. Carbon emissions for the zero-shot approach 312
A.6. Evaluation results of HubLink's retrieval performance for different question complexity types 313
A.7. Evaluation results of HubLink's retrieval performance for different use cases 314
A.8. Evaluation results of the retrieval performance for different use cases . . 315
A.9. Evaluation results of the retrieval performance for different question types 316
A.10. Evaluation results of the retrieval performance for different hop counts . 317

List of Acronyms

ACM Association for Computing Machinery.

AE Artifact Evaluation.

AI Artificial Intelligence.

ANN Approximate Nearest Neighbor.

BLEU Bilingual Evaluation Understudy.

CCS Computing Classification System.

CQs Competency Questions.

CS Computer Science.

DIKW Data, Information, Knowledge, Wisdom.

DL Deep Learning.

DOI Digital Object Identifier.

EASE International Conference on Evaluation and Assessment in Software Engineering.

EBSE Evidence-based Software Engineering.

ECSA European Conference on Software Architecture.

EM Exact Match.

ESEC/FSE ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

FAIR Findable, Accessible, Interoperable, Reusable.

FN False Negative.

FP False Positive.

GPU Graphics Processing Unit.

GQM Goal, Question, Metric.

IAA Inter-Annotator Agreement.

ICSA International Conference on Software Architecture.

ICSE International Conference on Software Engineering.

IEEE Institute of Electrical and Electronics Engineers.

KARAGEN Knowledge Augmented Retrieval And Generation ENgine.

KE Knowledge Engineering.

KG Knowledge Graph.

KGE Knowledge Graph Embedding.

KGQA Question Answering over Knowledge Graphs.

KM Knowledge Management.

KMS Knowledge Management System.

LLM Large Language Model.

MAP Mean Average Precision.

MDSD Model-Driven Software Development.

ML Machine Learning.

MRR Mean Reciprocal Rank.

NLI Natural Language Inference.

NLP Natural Language Processing.

NLU Natural Language Understanding.

OFAT One-Factor-at-a-Time.

ORKG Open Research Knowledge Graph.

QA Question Answering.

RAG Retrieval-Augmented Generation.

RAGAs Retrieval Augmented Generation Assessment.

RDF Resource Description Framework.

RDM Research Data Management.

RE Requirements Engineering.

RKG Research Knowledge Graph.

RLC Research Life Cycle.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

SDLC Software Development Life Cycle.

SE Software Engineering.

SPL Software Product Line.

SPLE Software Product Line Engineering.

SWA Software Architecture.

SWEBOK Software Engineering Body of Knowledge.

T5 Text-to-Text Transfer Transformer.

TN True Negative.

TP True Positive.

TtV Threats to Validity.

VCS Version Control System.

Part I.

Preliminaries

1. Introduction

In this doctoral thesis, we investigate the organization and structuring of research knowledge, as well as the application of semantic knowledge representations for Software Engineering (SE) research artifacts (e.g., scientific papers, corresponding datasets and software). We contribute, among other things, to meta-research in SE (i.e., research on research or the science of science (lato sensu), from Latin "scientia", meaning "knowledge"[1]) to support sustainable research knowledge description and knowledge access.

In the following, we discuss the research goals and main research question (Section 1.2), as well as envisioned solutions and contributions (Section 1.3) derived from the discussed problem statements (Section 1.1) to provide a research overview of this doctoral thesis (Section 1.4). Further, we identify roles and perspectives (Section 1.5) that might benefit from our approach in the long term. Finally, we conclude with a thesis outline in Section 1.6.

1.1. Background and Problem Statements

Software, as a technological artifact, plays a fundamental role in modern life by driving innovation and automation across scientific research and industrial practice. It is a key enabler of the digital transformation, highlighting software's pervasive impact on economic and technological development (cf. essay entitled "Why Software Is Eating The World"[2] by Marc Andreessen). The Software Development Life Cycle (SDLC) ensures through different development activities (i.e., planning, analysis, design, development, testing, deployment, and maintenance) that software meets user requirements, enables software quality, and can be maintained over time [244]. To this end, SE offers a systematic approach for designing, developing, operating, testing, documenting, and maintaining software (see IEEE Std 610.12-1990 [111] – updates IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology (ANSI) [110]). In this context, SE research investigates, supports, and improves, among other things, these activities, and provides new solution approaches. The effects of such research are reflected either in much higher product quality or in improved process characteristics. For instance, the automation of an existing (manual) process can lead to lower costs, faster product results with high quality, or even a reduction in the required expertise [130]. Thus, resulting findings in this research field serve as a key pillar for progress and innovation cycles in science, practice, and society while supporting the digital transformation. However, research outcomes in this field require effective science communication [29, 317] as a natural extension of Open Science

[1] According to the Oxford Latin Dictionary [Last accessed on 2025-10-03]

[2] Andreessen, Marc. "Why Software is Eating the World". Life & Culture. The Wall Street Journal. (August 20, 2011) https://www.wsj.com/articles/SB10001424053111903480904576512250915629460 [Last accessed on 2024-12-06]

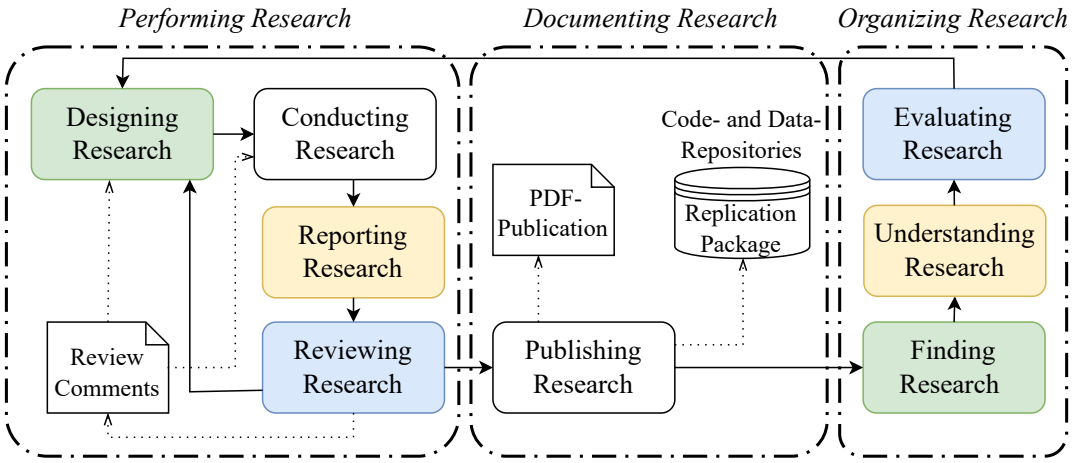

Figure 1.1.: Overview of typical phases and their dependencies in a research life cycle comprising different research activities and derived research artifacts such as papers and replication packages (Figure adapted from Kaplan [129])

and empirical evidence to achieve a breakthrough into practice. This indicates the maturity of a research field, which is investigated by evidence-based software engineering [150] and covered by replication artifacts [153] to ensure the reliability of research results.

Since the beginning of modern science, scientific papers have been the central medium and primary research artifact for communicating findings and insights that enhance collective knowledge and advance state-of-the-art. This practice is often referred to as the document-centric approach [13, 117]. The SE research community has adopted this procedure, since the very beginning of the first SE venue in 1968[3] (later known as International Conference on Software Engineering (ICSE) and one of the top-tier conferences). Those papers are also mostly aligned with replication artifacts, such as data and software. To create such artifacts, researchers face several activities (similarly to the SDLC, which is used to develop and maintain software) within an Research Life Cycle (RLC) (Figure 1.1):

(i) *Designing Research*: Researchers define research questions or hypotheses and have to select appropriate methodologies to ensure validity and evidence.

(ii) *Conducting Research*: Researchers have to operationalize the research design and carefully document intermediate research steps and results based on the research questions.

(iii) *Reporting Research*: Researchers have to communicate research results and findings in an understandable, comparable, and trustworthy way. This includes writing research papers, creating presentations, and providing replication artifacts, if possible.

(iv) *Reviewing Research*: Researchers have to objectively assess the research work of other researchers (peer review) and understand how to evaluate research. Peer reviews help to maintain the quality and integrity of scientific findings.

(v) *Publishing Research*: Researchers should aim to provide their research artifacts (paper, software, data) through open access over the long term for transparency, trust, and reuse.

[3] The NATO Software Engineering Conference: `http://homepages.cs.ncl.ac.uk/brian.randell/NATO/` [Last accessed on 2024-12-06]

(vi) *Finding Research*: Researchers have to identify related work and research gaps to justify the relevance and novelty of their own research and that of others.

(vii) *Understanding Research*: Researchers have to understand and summarize the most relevant parts and findings of a paper in a specific knowledge field. This involves critically analyzing the literature to extract key insights and trends, which can then be used to inform future research directions.

(viii) *Evaluating Research*: Researchers have to analyze and structure scientific contributions across documents. This process helps in creating an understanding of the state-of-the-art.

In the general RLC, these activities and the corresponding individual tasks can be aggregated into three main areas: *performing research*, *documenting*, and *organizing research*, as depicted in Figure 1.1. The activities in each area have a counterpart, as respectively colored in the figure. For instance, we can support a better understanding of research, if we have a better understanding of how we report scientific contributions in research artifacts. As in SE research, meta-research aims to improve the overall process and research activities by identifying and addressing issues, and, thus, promoting and developing best practices, conceptual approaches, and tool support while ensuring quality (cf. Figure 2.1).

One of the current major weaknesses associated with the presented RLC in SE concerns scientific publications, which face two key issues. First, scientific publications are typically written in natural language and disseminated as monolithic `PDF` documents, which are inherently semi-structured to unstructured. This lack of formalization limits machine-readability and hinders automated processing, interpretation, and reuse across the aforementioned research activities. Second, this document-centric workflow often fails to meet the Findable, Accessible, Interoperable, Reusable (FAIR) principles [306, 13, 117, 109]. As a result, little effort in SE research is invested in structured knowledge classification and representation. Semantic publishing approaches that use machine-readable formats, such as the Resource Description Framework (RDF)[4], and the corresponding adoption of Research Knowledge Graphs (RKGs), such as the Open Research Knowledge Graph (ORKG)[5] [109] (cf. Section 2.6), offer promising solutions. These approaches support sustainable Research Data Management (RDM) [237] and enable a shift from document-centered to knowledge-centered scholarly communication.

To support this transition, it is essential to define relevant (descriptive) metadata that captures the structure and context of research artifacts across the RLC. However, this is particularly challenging in SE, where research outputs vary widely, from empirical studies and solution proposals to experience reports and opinion papers, for which no common understanding exists [29, 273]. Thus, the SE community lacks a unified classification schema that enables consistent comparison and integration of research contributions. Existing classifications are fragmented and vary in terminology and scope. A comprehensive and general classification concept that can be refined and instantiated for different SE research subfields, such as Software Architecture (SWA), is needed to support machine-readability of research artifacts. As research practices evolve, a maintenance and general research approach to cope with refined community understandings and changes over time is crucial.

[4] `https://www.w3.org/RDF/` [Last accessed on 2025-10-03]

[5] `https://orkg.org/` [Last accessed on 2025-10-03]

Problem Statement P1 w.r.t. *performing and documenting research*: Classification schemas and concepts are used to describe the scientific knowledge statements of SE research outcomes and the content of SE papers in general (e.g., distinguishing methods, reports, reproducibility, and evaluation in a knowledge field). Regarding the consolidation of the large body of classified knowledge in SE research, a generally applicable and maintainable classification schema is crucial to provide a common, holistic, and long-lasting understanding. However, the commonalities and differences as well as knowledge gaps among different classification schemas have rarely been studied. Currently, there is a lack of conceptual approaches in which classification concepts are central and are technically supported by appropriate machine-readable formats for effective documentation and analysis. As they are subject to change, we need an evaluation approach to cope with their evolution.

Scientific papers remain the primary medium for communicating and reviewing research findings in SE [29] and other research fields. The current state-of-practice for accessing scientific literature has evolved in parallel with the development of the World Wide Web. To this day, researchers still have to rely on scientific search engines (cf. Section 3.2), such as digital libraries and indexing services, which predominantly disseminate research knowledge as digital PDF documents.

To conduct literature searches using these platforms, researchers require advanced information literacy skills [290] (cf. Section 3.2.2). This means that they need to identify relevant keywords and to construct an effective search query via Boolean operators to broaden or narrow the number of the search engine results. Even for domain experts in SE, it is difficult to keep track of relevant keywords, especially in such a fast-evolving research field as SE and given the overwhelming publication flood (cf. Garousi et al. [75]). During this iterative procedure, researchers are dealing with unclear semantic relations, such as synonyms (words with similar meanings), hyponyms (hierarchical relation where a word is a subtype of another) and hypernyms (the inverse of hyponyms), and homonyms (same spelling but different meanings). These linguistic challenges hinder precise retrieval and often result in suboptimal search outcomes. Thus, finding an optimal query for an inquiry or search request is a time-consuming task and an error-prone step [330]. Moreover, current investigations have revealed shallow search features of scientific search engines in SE and a considerable gap between the search needs of researchers and the usability (cf. Li and Rainer [182]). Once a query is formulated and compiled, the search engines deliver an overwhelming number of sequentially listed papers with the title, a few text snippets, and a linked PDF for manual inspection. This leads to information overload, making it particularly challenging to synthesize and compare findings across several papers. Missing relevant research leads to (partial or full) re-inventions or to actually foreseeable failing research projects when existing relevant and negative results are unknown.

In summary, these current search practices, primarily based on keyword queries in academic search engines and PDF-based access, are increasingly inadequate for navigating the immensely growing volume of scientific SE literature. The reliance on unstructured text and limited metadata hampers advanced computational analysis and retrieval. Researchers face significant cognitive and procedural burdens when identifying relevant work, especially in tasks such as systematic literature reviews, related work discovery, or consensus

building [252]. Search processes often involve multiple sessions with vague semantic matches and iterative query refinement. Moreover, the lack of structured knowledge representation leads to inefficient filtering and frequent oversight of relevant contributions, including negative results and research gaps. This contributes to redundant research efforts and missed opportunities for knowledge and research artifact reuse.

Despite the existence of a broad spectrum of conceptual models for SE research knowledge classification (see problem statement 1) and supporting machine-readable and FAIR-compliant infrastructures, their integration into AI-driven retrieval systems remains limited. As research terminologies and knowledge structures continue to evolve, there is a pressing need for schema-agnostic, training-free retrieval approaches that can flexibly adapt to diverse and dynamic scientific knowledge representations.

Problem Statement P2 w.r.t. *documenting and organizing research*: Scientific search engines, which are commonly used to access software engineering (SE) research in the form of `PDF` publications, require researchers (domain experts) to possess advanced information literacy skills and are increasingly strained by information overload due to the exponential growth of SE publications. They are restricted to keyword-based searches and offer only basic filtering options (e.g., bibliographic metadata like publication year or author names). This hinders the application of semantic search techniques and impedes efficient knowledge discovery. Consequently, current SE practices invest little effort in effective and reliable classification, evolving representation (see (P1)) and storage, and retrieval (access) of SE research knowledge.

1.2. Research Goal and Main Research Question

In the following section, we describe the long-term vision and derive the corresponding research goals G1 and G2 to address the research problems P1 and P2 discussed in the previous section. We aim to support dedicated research activities within the RLC that are assigned to the main areas of *performing research*, *documenting research*, and *organizing research* by describing, persisting, and accessing research knowledge based on suitable metadata classification schemas that enable the full potential of digitalization.

Long-term Vision Development of a sustainable research knowledge and data management for software engineering research and its research subfields (e.g., software architecture research) to unlock the full potential of digitalization and, thus, enable machine-actionability.

This vision mainly originates from the position statement by Berners-Lee and Hendler in 2001 that is based on the semantic web activities[6], run by the World Wide Web consortium (W3C). In their publication, they state *"the Semantic Web will likely profoundly change the very nature of how scientific knowledge is produced and shared, in ways that we can now barely*

[6] `https://www.w3.org/2001/sw/`, [Last accessed 2025-01-30]

imagine" [28]. This statement is embedded in the late stage of the 5th Kontradieff wave[7], i.e., the age of information and telecommunications [200], and emphasizes the need for machine-readable and carefully structured, well-designed metadata for research artifacts so that machines can perform well-defined operations on them and, thus, contribute to machine-actionability and Artificial Intelligence (AI) readiness [237].

Based on the long-term vision, we derive two subsequent research goals specified for this doctoral thesis in the following:

Research Goal G1: Knowledge Description The development of a maintainable classification approach, along with a common foundation for knowledge specification and representation, is essential for accurately capturing the nature of SE research. This classification schema acts as a backbone, enabling to focus on and derive domain-specific concerns in SE research subfields, such as software architecture research. Ultimately, the goal is to enable the documentation of SE research knowledge that supports transparency, comparability, and reuse across the RLC. This systematic management of acquired SE knowledge supports the aforementioned research activities by providing machine-readable formats. Furthermore, a maintainable knowledge description approach facilitates sustainable research knowledge and data management. It unlocks the full potential of digitalization, enabling effective and efficient knowledge access, thereby supporting Research Goal G2.

Research Goal G2: Knowledge Access The implementation and specification of machine-processable formats for the developed classification and data schemas within a FAIR infrastructure are crucial for enabling effective knowledge retrieval. Ultimately, the goal is to build a knowledge base grounded in the concepts outlined in Research Goal G1. This derived knowledge base should be constructed and maintained semi-automatically (i.e., knowledge graph population) and form the basis for a schema-agnostic knowledge retrieval approach capable of adapting to evolving classification schemas. In doing so, we aim to explore the capabilities of AI4Research / AI4Science[8], specifically addressing knowledge access.

To achieve the research goals, the aforementioned problems have to be addressed. Referring to these problems, research questions are formulated and subsequently introduced in Section 1.4. To bridge both research problems and research goals, we derive one central research question (RQ):

RQ: How can we map contribution types and validated statements to research questions?

Regarding our first research goal, we interpret this question to focus on metadata for a classification approach describing research knowledge and artifacts in SE. Regarding

[7] According to the theory of Kondratieff [152] capitalist economies experience recurring periods of high growth followed by stagnation or decline, typically lasting 40 to 60 years. Each wave is driven by a cluster of technological innovations that fundamentally reshape production, consumption, and societal structures.

[8] Note: Similar to AI4SE (AI for Software Engineering), AI4Research / AI4Science refers to the application of AI techniques to accelerate and enhance research processes and activities in the RLC.

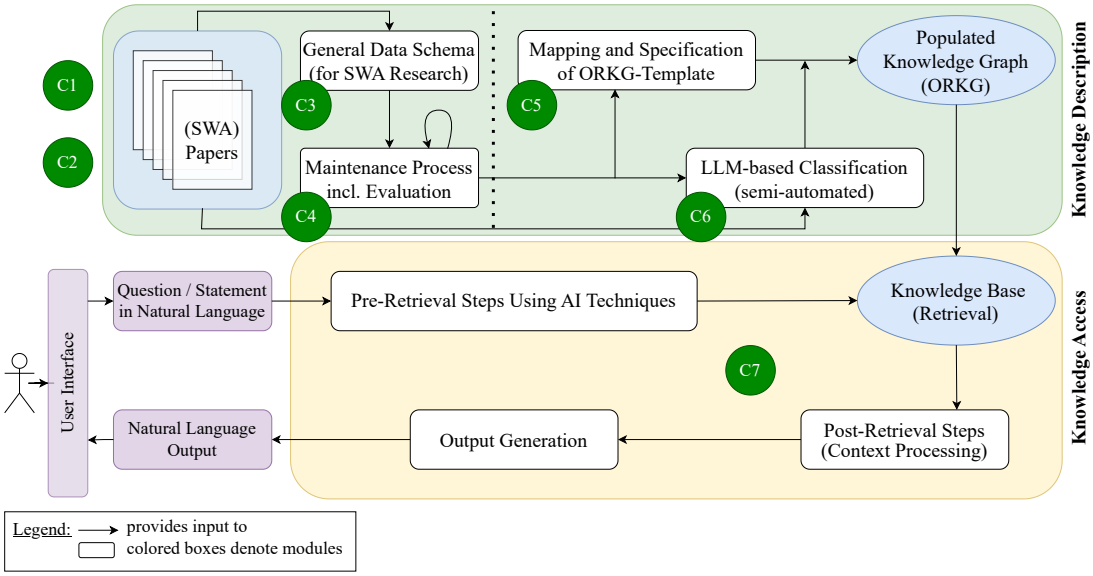

Figure 1.2.: Overview of our approach Knowledge Augmented Retrieval And Generation ENgine (KARAGEN) and the corresponding contributions (C1–C7) within the knowledge description module (green box) and the knowledge access module (purple and yellow boxes) (Figure extended from Kaplan et al. [133])

our second research goal, we refer to the corresponding question-answering dataset and taxonomy for the knowledge retrieval evaluation. In this way, we can bridge both research goals analogously on a meta-level.

1.3. Envisioned Solution and Contributions

In the following section, we describe the envisioned solutions and their respective contributions in the knowledge description and the knowledge access modules, as depicted in Figure 1.2.

C Our overall approach, KARAGEN (cf. Figure 1.2), comprising the *research knowledge description* module (green box) and the *research knowledge access* module (purple and yellow boxes).

Our overall approach (C) transforms the central research question in Section 1.2 into a more concrete technical and practical research question (RQ'):

RQ': How to construct a solution approach that combines the advantages of knowledge graphs and large language models to ease (i) knowledge description and (ii) knowledge access?

As a result, we first derive four contributions for an SE research knowledge description approach on a conceptual level:

C1 A novel *evaluation approach* for hierarchical classification and data schemas based on the Goal, Question, Metric (GQM) paradigm (cf. Section 2.8.1) w.r.t. the consecutive evaluation goals (i) *structure's suitability*, (ii) *applicability*, and (iii) *purpose* with cross-cutting evaluation strategy dimensions. This also includes the specification of *sustainability metrics* in terms of performance and energy consumption for machine-actionable solutions in software-intensive systems.

C2 A novel *documentation process for sustainable literature studies* in the data extraction phase, especially in large-scale reviewer contexts (data extractors), enhancing current and established conceptual guidelines for systematic literature studies in SE research (see Kitchenham and Charters [147]).

C3 A novel *classification and data schema* for describing research artifacts according to their findings, validity, and evidence, instantiated for SWA research with a resulting gold standard dataset in this research field and a corresponding visualization tool.

C4 A *maintenance and general research process* for the knowledge description of SE research artifacts (i.e., papers and replication artifacts) by analyzing existing or emerging approaches.

Building on the previously introduced conceptual contributions, we derive additional contributions aimed at supporting the development and evolution of machine-readable formats within a FAIR-compliant infrastructure. This involves enabling semi-automated Knowledge Graph (KG) population and maintenance of a research knowledge base. This knowledge base serves as the foundation for the retrieval module, enabling access to research knowledge. In our experiments, we use the ORKG infrastructure.

C5 A documented transformation from the initial classification and data schema (C3) into machine-readable and FAIR-compliant formats through the creation of *ORKG template specifications*. This includes a discussion of modeling alternatives and design decisions to support structured knowledge representation and KG population.

C6 A *classification framework* for deriving and benchmarking Large Language Model (LLM)-based classifiers for research artifacts and evolving schemas based on contrasting strategies, i.e., transfer learning and prompting, to enable semi-automated KG population and literature updates based on (evolving) metadata schemas.

Finally, we conclude with the contribution of our knowledge access module.

C7 A *retrieval framework* to develop and benchmark retrieval approaches based on different knowledge bases (i.e., embeddings / chroma vector store[9] (open-source) and RDF-based KGs). This framework includes our *novel schema-agnostic retrieval approach*, Hublink, embedded in a graph-based RAG-inspired pipeline.

In addition to the main research questions (RQ and RQ'), we define corresponding subordinate research questions (cf. Section 1.4) that align with the specific contributions presented (C1–C7). Each of these questions is associated with dedicated validation questions, which are introduced and discussed in the respective sections of the dissertation.

[9] `https://www.trychroma.com/` [Last accessed on 2025-10-03]

1.4. Research Overview

Figure 1.3 depicts the relations between our research goals supporting the long-term vision (Section 1.2), the identified problem statements (Section 1.1), the derived research questions (RQs) and corresponding contributions (Section 1.3) of our approach KARAGEN. The main research question (RQ) bridges the central modules of KARAGEN: *knowledge description*, which involves the selection and organization of metadata to characterize research artifacts by a (maintained) knowledge description approach, and *knowledge access*, which encompasses the retrieval of relevant information (based on metadata) through a question-answering system. In the following section, we introduce the research questions for each research goal, aligned with the problem statements, contributions, and evaluations to provide a research overview.

Research Goal G1 Addressing the first research goal G1 and its corresponding problem statement P1 requires conceptual solutions that support the systematic development, documentation, and maintenance of knowledge description approaches for research artifacts. To this end, we address research questions RQ I-1, RQ I-2, RQ I-3, and RQ I-4 by establishing a foundational understanding of how such approaches can be constructed, evaluated, and sustained over time. This allows bridging different SE communities and their various understandings and definitions of metadata. In the following, we subsequently introduce the specific research questions associated with (G1):

RQ I-1: How to evaluate (hierarchical) classification schemas as a knowledge description approach (in software engineering research), and how to guide researchers through such an evaluation?

RQ I-2: How to support open science principles in literature studies to ensure transparency and replicability in the data extraction process, especially in a large-scale reviewer context, and how to manage such a process?

With these research questions, we build the methodological foundations for this thesis. First, we introduce our evaluation approach (C1)[10] with three sequential process evaluation steps for research knowledge description approaches. This contribution allows evaluating current constructed knowledge description approaches and evolution scenarios to support maintainability systematically by unifying and harmonizing evolving classification concepts (see contribution C4). A documentation process to trace the intermediate evaluation steps is proposed by contribution C2. The corresponding guidelines can also be conducted in a large-scale reviewer (data extractor) context (cf. Chapter 7). We show the feasibility of our approach using a running example (Section 5.2.1) and use case examples by contributions C3 (Section 7.5), C4 (Section 8.3), C6 (Section 12.4.2), and C7 (Section 13.4.2). In addition, we define quality criteria for the experimental setup in Section 10.2, important for the evaluation of contributions C6 and C7. Further, we conduct a citation analysis to investigate the short-term impact of our approach (Section 5.4).

[10] Side note: We can use the evaluation approach for self-validation as well, as the GQM approach inherently follows a hierarchical structure comprising goals, properties, and metrics. This allows researchers to engineer our proposed evaluation approach from a meta-research perspective (cf. Section 2.1.1).

RQ I-3: How to describe and classify research artifacts w.r.t. their findings, validity, and evidence (in software architecture research)?

With this research question, we aim to establish a research knowledge description approach (C3), beginning with the core classification concepts. Specifically, we propose a schema for characterizing research artifacts, such as scientific papers and supplementary materials, based on their research findings, which reflect advancements in the state-of-the-art, and their validity, which indicates the strength of supporting evidence. To demonstrate the feasibility of this approach, we instantiate the schema within a domain-specific area of SE, namely SWA research. The schema is evaluated using our proposed evaluation framework (C1) and further validated through a literature review of papers from top-tier SWA venues (Section 7.7 and Section 7.8).

RQ I-4: How can classification and data schemas for knowledge description of software engineering research artifacts be maintained over time to accommodate evolving research practices and metadata requirements?

With this research question, we aim to construct a general research process to build and evaluate knowledge description approaches and support evolution scenarios w.r.t. classification concepts and concrete instances. We use this approach to point out commonalities and differences between these existing classifications, while analyzing research classification gaps. Further, we use our documentation process regarding data extraction (C2) of included SE meta-research classifications. The results from our literature review serve as a reusable dataset for constructing a classification schema to structure the body of SE research knowledge according to their research artifacts and corresponding (descriptive) metadata.

Research Goal G2 Addressing the second research goal G2 and its corresponding problem statement P2 requires the transformation of the conceptual knowledge description approaches developed under (G1) into broadly standardized machine-readable formats (C5), embedded within a FAIR-compliant infrastructure. This enables machine-actionable knowledge retrieval via the access module (C7). To promote this, we introduce a development and evaluation platform that facilitates to derive classifiers to enable semi-automated knowledge graph population (C6) and adaptive knowledge access (C7), capable of handling evolving classification schemas. Additionally, the platform supports tracking energy consumption and performance metrics, enabling informed trade-off decisions between efficiency and sustainability. These efforts aim to address research questions RQ II-1, RQ II-2, and RQ II-3. Following, we subsequently introduce the specific research questions associated with (G2):

RQ II-1: How to transfer and document our resulting data schemas to the concepts of ORKG templates? What are (trade-off) design decisions?

With this research question, we aim to transfer our conceptual classification and data schema to the concepts of ORKG templates as RDF-based and FAIR-compliant formats.

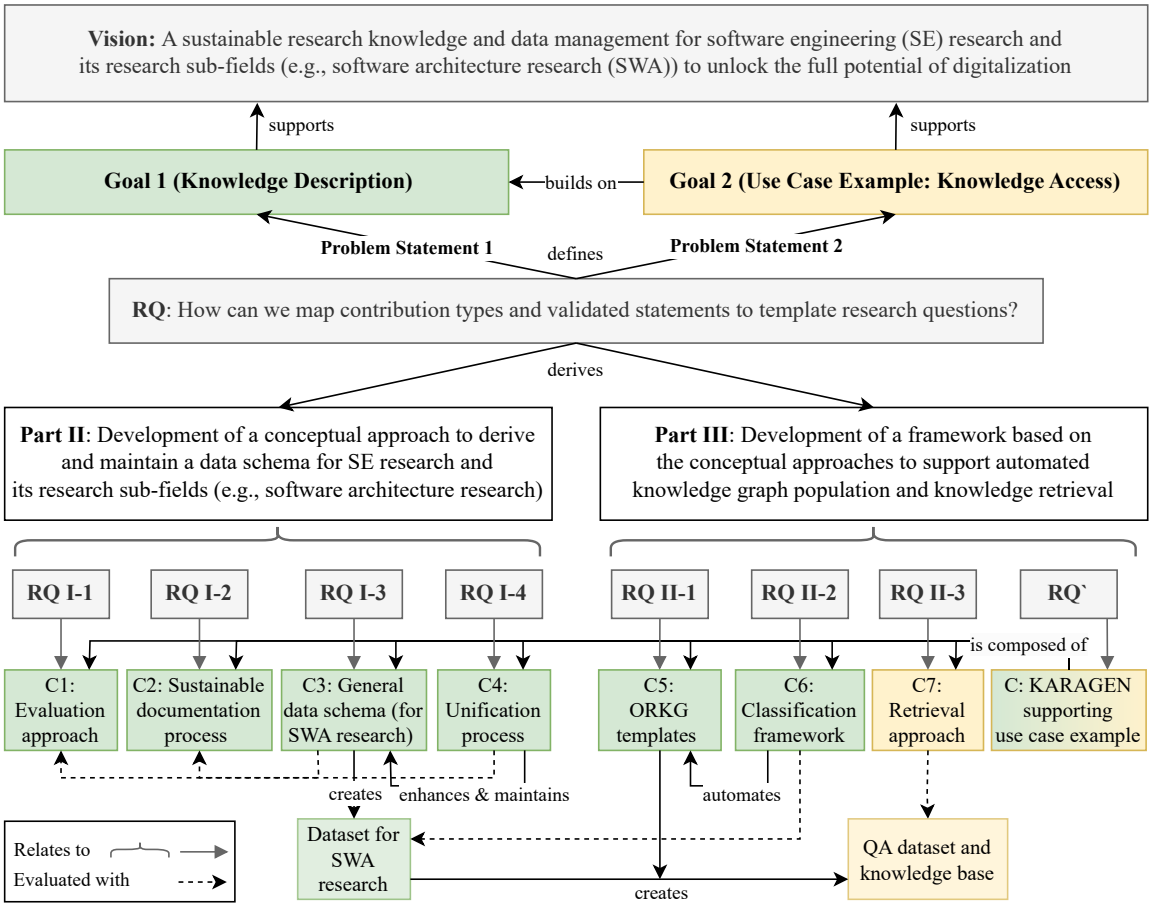

Figure 1.3.: Relation between long-term vision, main research goals, the main research question RQ, and corresponding contributions C1–C7 (Figure extended and adapted from Kaplan [129])

We intend to create the domain-specific ORKG template for SWA research in such a way that it provides long-lasting value for prospective use w.r.t. the characteristics: (i) maintainability, (ii) usability (i.e., readability and understandability of templates for users), and (iii) suitability for the intended purpose (e.g., knowledge retrieval). This includes a discussion of design decisions w.r.t. the different modeling options in ORKG to model state-of-practice and normative aspects. Furthermore, we derive additional three variants to this data schema version to evaluate the robustness of our schema-agnostic retrieval approach (see contribution C7).

RQ II-2: How to construct a semi-automated classification framework using natural language processing techniques for classifying research artifacts?

With this research question, we address the problem on how to automate the classification task for the derived (hierarchical) data schemas in (C3) and (C4) using parts of (C1) for evaluation, focusing on performance and sustainability aspects of the different classification approaches (based on fine-tuning and prompting). As this is a complex task and many challenges exist, we build a framework to cope with these challenges. This way,

we are able to conduct the KG population in a semi-automated way to build a knowledge base for our retrieval approach, thereby supporting scalability and efficiency.

RQ II-3: How can the limitations of schema-dependent and training-based Question Answering over Knowledge Graphs (KGQA) systems in scholarly domains be overcome by integrating research knowledge graphs with large language models for schema-agnostic and provenance-aware retrieval?

With this research question, we aim to introduce our novel retrieval approach in a development and evaluation framework for benchmarking and reuse. SE research knowledge, based on the aforementioned presented concepts and the developed knowledge database for SWA research, enables a novel form of scholarly communication. This is in contrast to the current paper-based and keyword search approach for knowledge documentation and retrieval. As classification and data schemas may vary due to domain-specific adaptions, refinements, and ongoing evolution, we design our retrieval approach schema-agnostic and training-free. Furthermore, our retrieval approach also aims to complement the advantages of KGs and LLMs to enhance state-of-the art and state-of-practice. Based on a competency question taxonomy (cf. Section 10.3) and parts of our evaluation approach (C1), we evaluate our approach regarding retrieval and answer generation quality. Answering RQ II-3 helps to uniformly and systematically manage SE research knowledge and to support finding, understanding, and evaluating SE research (cf. respective research activities of a RLC in Figure 1.1).

1.5. Roles and Perspectives

The contributions of the knowledge description and knowledge access modules that we construct and develop in this doctoral thesis may provide several benefits, as detailed in Section 15.2, for the following roles and derived perspectives, according to the main activities of a representative of a role, in a scientific ecosystem:

- *Author, planning and designing a research project (prospective)*: An author who plans and organizes a research project has to structure each research step by defining a corresponding research design and suitable research questions to operationalize the method. For communication and collaboration, it is helpful to rely on a common foundation for the description of research artifacts. To justify the relevance of the own work, an overview of current related work and research trends is needed.

- *Author, reflecting on a research project (retrospective)*: An author who critically reflects on own work of a former research project to analyze and verify the research design w.r.t. current research practices to enhance own research skills and competencies.

- *Reviewer, assessing a research project*: A reviewer who aims to evaluate a research project in a holistic, transparent, and comparable manner, with the objective of assessing both the relevance and originality of the research object under investigation, as well as the appropriateness of the corresponding method design.

- *Evidence-based Researchers, delving into SE research*: A researcher that contributes to evidence-based research in SE [150] by having a supporting classification tool to position the research contribution and justifying relevance of the research project.
- *Steering Committee and Program Chairs, defining settings for a venue while evolving a research community:* Members of steering committees and program chairs are responsible for shaping the thematic and methodological direction of a research venue and the research community. This includes setting criteria for submissions, identifying emerging research trends, and fostering coherence within the community.
- *Editors (especially, editors-in-chief as head of an editorial board), taking responsibility for the definition and implementation of the scientific strategy of a journal:* Editors play a critical role in ensuring the integrity, coherence, and relevance of published research. They require comprehensive and structured overviews of submissions to identify contributions that align with the journal's (or conference's) scope and maintain high editorial standards.
- *Replicator, repeating a research project*: A replicator who reads a publication comprising a replication artifact with the intent and understanding of repeating the empirical study or a family of experiments.
- *Practitioner, consulting and decision-making*: A practitioner who wants to assess the relevance of scientific research outcomes and contributions in order to determine the applicability of given technologies for a specific project context and set of constraints.
- *Knowledge Engineer and Developer, organizing and searching for relevant research*: A knowledge engineer who requires advanced tooling to benchmark AI-driven classification techniques and retrieval strategies for research artifacts, with the goal of developing domain-specific expert systems.

Each role within the scientific ecosystem engages with research knowledge in distinct yet interdependent ways. Across the full spectrum of activities in the RLC, from planning and assessing to publishing and reusing research artifacts (see Figure 1.1), the lack of structured, accessible, and maintained (domain-specific) knowledge representations remains a common barrier. Addressing these challenges through standardized and maintained, semantically enriched approaches and frameworks holds the potential to enhance transparency, knowledge accessibility, and sustainable engineering practices across the entire RLC (cf. Figure 2.1 in Section 2.1.1). This requires a better and deeper understanding of maintainable knowledge representation and effective access to research knowledge, as addressed in this dissertation.

1.6. Thesis Outline

The remainder of this doctoral thesis is structured into four main parts, as presented in the following:

Part I Preliminaries The first part introduces the preliminaries of this doctoral thesis. In the following Chapter 2, we provide foundations comprising relevant terms that are necessary for the understanding of the remainder. This also includes evaluation preliminaries for our evaluation approach and experiments in the contribution parts (Part II and Part III). In Chapter 3, we present current state-of-practice in SE research w.r.t. the research goals of this thesis, i.e., (i) research knowledge description and (ii) knowledge access.

Part II Knowledge Description in Software Engineering Research The second part focuses on the methodological and conceptual approaches and forms the basis for the third part (see below) to build an evolving knowledge base. An overview of this part is presented in Chapter 4. First, we introduce an evaluation approach (C1) using the GQM paradigm based on three sequential steps (goals) for (hierarchical) knowledge description approaches in Chapter 5. This approach is applied to evaluate our contributions in Part II and Part III. As part of this approach, we propose novel sustainability metrics that integrate performance and energy consumption considerations. This enables an assessment of the extent to which our solutions support Green AI principles. In Chapter 6, we introduce the documentation process for literature reviews, especially in a large-scale reviewer context, supporting the data extraction phase (C2). These chapters build the foundation for our novel introduced classification and data schema based on a general classification concept that we instantiate for SWA research (C3) in Chapter 7. To cope with evolving schemas, we present a general research process (C4) in Chapter 8.

Part III Knowledge Access in Software Engineering Research The third part focuses on practical and technical aspects based on the conceptual approaches of Part II. An overview of this part is presented in Chapter 9. First, we provide an overview of our overall approach, entitled KARAGEN (C) in Chapter 10. We set this approach in context of the use cases, we aim to support; and present evaluation preliminaries in terms of best practices for experimental studies using LLMs. Moreover, we present our question taxonomy (mapping contribution to (research) questions) to derive competency questions and build a holistic framework that can provide hints for the capabilities of a retrieval approach. Following in Chapter 11, we mainly present the ORKG templates (C5) for our SWA data schema (Chapter 7) as a case example and to build a knowledge base. The remaining chapters in this part encompass the developer and evaluation frameworks (C6, C7) for (i) deriving an automated and benchmarked LLM-based classifier to categorize papers according to their metadata to, e.g., populate a RKG (Chapter 12), and (ii) evaluating competing retrieval approaches in a Retrieval-Augmented Generation (RAG) pipeline (Chapter 13).

Part IV Reflections The final part of this thesis reflects and discusses the previously presented and evaluated contributions. We analyze related work in Chapter 14. In Chapter 15, we conclude with key findings by summarizing the research questions for each contribution. Furthermore, we discuss the relevance and implications of our main contributions for knowledge description and access, considering diverse perspectives, roles, and stakeholders (cf. Section 1.5). The chapter closes with an outlook on future research directions building upon the presented work.

2. Foundations

In this chapter, we present foundations of this thesis and introduce the terminology and the concepts that are conducive to a better understanding of the remainder. First, we introduce relevant scientific terms in Section 2.1. Next, we present relevant concepts in machine learning (Section 2.2) and natural language processing and understanding (Section 2.3). Following, we discuss the organization of knowledge (Section 2.4) and different modeling concepts for knowledge representations (Section 2.5). Based on this, we introduce the ORKG (Section 2.6) that we use as infrastructure for our proposed solutions. Underlying our conceptions and contribution in this doctoral thesis, we clarify terms in the context of sustainable software development and Green AI (Section 2.7) in times of resource-intensive LLMs. Finally, we conclude this chapter with evaluation preliminaries (Section 2.8).

2.1. Terminology

In this section, we introduce relevant terms based on standards, scientific literature, and our own considerations in the context of Software Engineering (SE) and SE research.

2.1.1. Scientific Ecosystem and Meta-Research

The ***Scientific Ecosystem*** encompasses the complex, interdependent network of researchers, academic institutions, funding bodies, policymakers, industry partners, and other stakeholders engaged in the generation, validation, and dissemination of scientific knowledge. This system functions through dynamic interactions among its constituents, with the exchange of information, resources, and expertise driving both incremental and transformative advances in understanding. Comprehensive analysis of the scientific ecosystem is crucial for optimizing research processes, fostering collaboration, and ensuring that scientific outputs are robust, reproducible, and responsive to societal needs. By aligning the ecosystem's structures and incentives with stakeholder objectives, it is possible to enhance both the efficiency and the long-term impact of scientific inquiry [93].

In doing so, ***Meta-Research***, also known as ***Metascience***[1] [222], is an emerging scientific discipline that aims to achieve evidence-based improvements in research and research practices to contribute to an improved scientific ecosystem. The strands of metascience represent an intersection of Open Science (cf. Section 2.1.3), Science of Science, and Methodological Activism [222]. This discipline uses, among other things, theoretical and empirical investigations (e.g., meta-analyses, surveys, or interviews) to explore how research is conducted and what improvements can be made to build solutions for identified problems. For this purpose, Hardwicke et al. [93] proposed a translational framework

[1] `https://www.cos.io/metascience-alliance` [Last accessed on 2025-08-14]

Figure 2.1.: The translational framework for meta-research acc. to Hardwicke et al. [93]

in 2020 to guide meta-research efforts through four sequential stages: (a) identifying problems, (b) investigating problems, (c) developing solutions, and (d) evaluating solutions, as depicted in Figure 2.1. As the scientific ecosystem continues to evolve, the discipline of meta-research presents an opportunity to use empirical evidence to guide its development and maximize its potential. According to Ioannidis et al. [112], meta-research is "taking a bird's eye view of science" and comprises five main topic areas that we describe in the following:

1. **Methods** ("How to perform research?"): This topic area addresses aspects such as study design, methodological rigor, statistical analysis, research synthesis, collaboration practices, and ethical considerations.

2. **Reporting** ("How to communicate research?"): This topic area focuses on reporting standards, study registration, disclosure of conflicts of interest, and the effective communication of research findings to various stakeholder groups (e.g., SE practitioners, educators, and researchers).

3. **Reproducibility** ("How to verify research?"): This topic area includes the sharing of data and methods, and emphasizes repeatability, replicability, reproducibility, and mechanisms for self-correction within the scientific process.

4. **Evaluation** ("How to evaluate research?"): This topic area includes both pre- and post-publication peer review, criteria for research funding, and other mechanisms for evaluating the quality and impact of scientific work.

5. **Incentives** ("How to reward science?"): This topic area investigates the reward systems embedded within research institutions, including promotion criteria, rewards and penalties in research evaluation for individuals, teams and institutions.

By addressing these topic areas, meta-research contributes to the refinement of the scientific ecosystem. It provides an empirical basis for designing, implementing, and evaluating reform initiatives aimed at elevating research standards. From a software engineering perspective, we are able to contribute with constructive solutions to support different scientific tasks conducted in the aforementioned meta-research topic areas and to develop solutions, as indicated in the translational framework. Consequently, these principles lay the foundation for this doctoral thesis.

2.1.2. Software Engineering Research Terminology

SE research is, at its core, of a constructive nature, as it provides solutions (e.g., technological solutions) to given problem specifications (see Wieringa et al. [305]). In the following, we introduce the terms and definitions used throughout this doctoral thesis.

Definition 2.1.1 (Research Artifact)**.** A (digital) research artifact represents all relevant concrete and abstract artifacts, i.e., all relevant data, descriptions, and information related to a self-contained research project (cf. Definition 2.1.3). We differ between papers, media (e.g., videos of paper presentations or presentation slides of a paper), and replication artifacts (software and data) as typical *Research Artifacts*.

Definition 2.1.2 (Metadata)**.** Metadata ("data that describes other data" [70]) of research artifacts refers to structured information that describes, locates, or otherwise makes it easier to retrieve, use, or manage research artifacts. The purpose of metadata may vary from finding artifacts (via search engines or repositories), understanding their context and contents, reusing them in future research, to verify and replicate the findings of research artifacts. Common types of metadata for research artifacts can be classified as:

(i) descriptive metadata (i.e., mainly helping to identify and discover research artifacts),

(ii) structural metadata (i.e., explaining how data elements are organized, e.g., dataset schemas, modular software architectures, or relationships between files),

(iii) administrative metadata (i.e., used for managing research artifacts),

(iv) technical metadata (i.e., describing the technical characteristics such as file formats, file sizes, or software dependencies),

(v) provenance metadata (i.e., describing the origin and history, e.g., who created or modified the data), and

(vi) access/usage metadata (i.e., providing information about access rights and usage metrics).

Note: In this doctoral thesis, we mainly focus on descriptive metadata.

Definition 2.1.3 (Research Project)**.** A project is any endeavor undertaken individually or in collaboration that involves the design and conduct of research steps (cf. Definition 2.1.9). It is carefully planned to achieve a specific research objective (cf. Definition 2.1.4) and is conducted within a specific time period with limited resources.

Definition 2.1.4 (Research Objective)**.** A research objective constitutes a (long-term) research goal (i.e., research aim) of a research project. To achieve this goal, researchers may define several sub-projects (work packages) and corresponding research steps (cf. Definition 2.1.9). In each step, a research method is applied.

Table 2.1.: Template statements for research objects and objectives

Empirical-based	**Solution-based**	**Evaluation-based**
to <investigative verb>	to <constructive verb>	to <evaluative verb>
to investigate	to design	to validate <property>
to examine	to construct	to assess <property>
to explore	to develop	to evaluate <property>
to review (literature data)	to implement	to compare with similar approaches
to collect / to construct	to propose	

Table 2.1 provides an overview of typical template statements for research objectives.

Definition 2.1.5 (Research Object)**.** A research object is the object under study. The object is investigated by applying a research method assigned to a dedicated methodology. A self-contained research project comprises at least one research object. We locate research objects in a problem domain (cf. Definition 2.1.11), i.e., all abstract artifacts to investigate in the real world (e.g., problem statements w.r.t. solution approaches, phenomena, research hypotheses, human-related issues, or historical insights based on literature data) and in the solution domain, including solution instances for a problem statement (cf. Definition 2.1.12)

Definition 2.1.6 (Research Question)**.** A research question frames a research project and operationalizes the concrete method design. A research question is always dedicated to a research object. Based on the scientific contribution (cf. Definition 2.1.10), we distinguish between (i) constructive questions in the sense of an engineering approach or a new solution approach, (ii) evaluative questions of the solution approach, and (iii) investigative questions in the problem domain.

Definition 2.1.7 (Methodology)**.** A methodology refers to the overarching research strategy and rationale of a research project or research step (cf. Definition 2.1.9). Methodologies aggregate similar research method types based on different evidence-based and data analysis approaches to generate research results. Each result, along with its corresponding research step, is concerned by threats to validity.

Definition 2.1.8 (Research Paradigm)**.** Research paradigms define the underlying approach and purpose of a study. In SE, five main paradigms guide how research knowledge is generated and applied:

(i) Descriptive: characterizes existing phenomena or practices without intervention (documenting how "it is").

(ii) Explorative: investigates underexplored areas to generate insights and hypotheses for future research.

(iii) Constructive: designs and implements new constructive artifacts (e.g., tools or methods) to solve identified problems.

(iv) Evaluative: assesses the properties or quality of existing artifacts, methods, or processes through empirical analysis.

(v) Normative: proposes standards, guidelines, or best practices based on prior findings to improve practice (documenting how "it should be").

Definition 2.1.9 (Research Step)**.** A research step is restricted by methodological considerations. A research step always corresponds to the application of a research method assigned to a methodology considering the research objective and the research domain (conventional SE topic, SE education and training, or SE meta-research). As an input characteristic of a research question or hypothesis and its associated research object, a research step represents both a procedural element of the investigation and the unit through which research outcomes are generated. These outcomes are expressed as research contributions, based on either knowledge-based or solution-oriented statements. A research project may consist of several research steps to be conducted in an iterative loop until the research objective (i.e., the research goal) is achieved.

Definition 2.1.10 (Contribution)**.** A contribution in software engineering updates or enhances the body of existing scientific or engineering knowledge. Consequently, we define a contribution in SE as

(i) a new solution to a dedicated problem as well as its update (new version),

(ii) evaluations of this solution approach and its results,

(iii) empirical investigations (i.e., primary research) and their outcomes in a problem domain, including syntheses of prior research (i.e., secondary and tertiary research), and

(iv) results of replication studies.

A research object (Definition 2.1.5) comprises one or more distinct contributions.

Definition 2.1.11 (Problem Domain)**.** A problem is a given situation that negatively impacts or disadvantages a human or technical entity (e.g., a researcher, practitioner, educator or software system). A problem domain aggregates several problem instances that an entity might face and represents the <real world>.

Definition 2.1.12 (Solution Domain)**.** A solution is a constructive instance to overcome a given problem (statement). This domain mainly represents the constructive nature of software engineering.

Definition 2.1.13 (Statement)**.** A statement (as a sentence, a part of a sentence, or as a semantically solid unit by aggregating sentences) is aligned with each contribution type. In this regard, we distinguish between solution-based and knowledge-based statements assigned to the different contribution types that may vary w.r.t. the validity aspects.

2.1.3. Open Science and FAIR Principles

The accessibility, transparency, and reusability of scientific results and knowledge are increasingly emphasized through concepts such as Open Science and the FAIR principles. Although they share the common goal of improving the dissemination and utility of scientific knowledge, they differ in scope, focus, and implementation.

Open Science (in SE) is a broad, interdisciplinary movement aimed at making all aspects of scientific research, including research artifacts and research activities, openly available and accessible to the public and the scientific community. It encompasses, but is not limited to, open access, open data, and open source. The primary objective is to foster collaboration, reproducibility, and equitable access to scientific knowledge, thereby accelerating innovation and societal impact [193].

In contrast, the **FAIR principles**, which stand for Findable, Accessible, Interoperable, and Reusable [306], provide a set of guidelines specifically designed to enhance the management and stewardship of digital research data. These principles focus on ensuring that data and associated metadata (cf. Definition 2.1.2) are structured in a way that facilitates discovery and reuse by both humans and computational systems. Key aspects include the use of persistent identifiers (abbreviated PIDs), standardized metadata schemas, interoperable formats, and clear licensing terms [306]. They have become a cornerstone of modern RDM and are actively promoted by initiatives such as the National Research Data Infrastructure (NFDI) and the European Open Science Cloud (EOSC) (see Section 3.3).

While both concepts promote openness and reusability, their distinction lies in their scope and technical orientation. Open Science is a cultural and procedural paradigm that advocates for openness across the entire RLC. The FAIR principles, on the other hand, are technically oriented and primarily concerned with the quality and usability of data assets. In summary, Open Science provides the overarching vision for transparent and inclusive research, whereas the FAIR principles offer a practical foundation for ensuring that research data are optimally managed and reused effectively. Together, they form complementary pillars for advancing responsible and effective scientific practice in the digital age.

2.2. From Artificial Intelligence to Generative AI

Artificial Intelligence (AI) is the broad and interdisciplinary field within computer science concerned with designing systems capable of performing tasks that, if executed by humans, would require intelligence. Such tasks include reasoning, learning, problem-solving, planning, perception, decision-making, and (natural) language understanding. AI encompasses a wide spectrum of approaches, ranging from symbolic reasoning and expert systems to statistical learning and neural computation. Despite various definitions of AI, a widely accepted definition by the High-Level Expert Group on Artificial Intelligence (AI HLEG) of the European Commission (EC) states that AI systems are "*Systems that display intelligent behaviour by analysing their environment and taking actions – with some degree of autonomy – to achieve specific goals*" [262, p. 16 and 20].

Machine Learning (ML) is a subfield of AI that focuses on algorithms and models that improve their performance on a task through experience, typically by learning patterns from data rather than being explicitly programmed. Unlike rule-based systems, ML models improve their performance through exposure to data, without being explicitly programmed for every task [203]. ML approaches are commonly categorized into supervised learning (learning from labeled datasets), unsupervised learning (discovering structure in unlabeled datasets), semi-supervised learning (hybrid approach), and reinforcement learning (learning optimal actions through trial-and-error interaction with an environment) [203]. ML serves as the foundation for many modern AI applications, including recommendation systems and Natural Language Processing (NLP) (see Section 2.3).

Deep Learning (DL) is a subset of ML that employs neural networks with multiple layers (deep architectures) to automatically learn hierarchical feature representations from raw data [172]. DL models, particularly those based on convolutional neural networks

(abbreviated CNNs), recurrent neural networks (abbreviated RNNs; based on concepts by John Hopfield (1982) and by Rumelhart (1986)) [50], and Transformer architectures (see Section 2.3.1), have achieved state-of-the-art results in domains such as computer vision and natural language processing. Their ability to jointly learn representation and decision-making pipelines has reduced reliance on manual feature engineering.

Generative Artificial Intelligence (GenAI) refers to models that are instantiated with a machine learning architecture (e.g., a deep neural network), and can therefore produce new data resembling the training data. This includes content such as text, images, audio, and more. The most impactful generative models are based on Transformer architectures (see Section 2.3.1), such as GPT (Generative Pre-trained Transformer).

The **relationship** between the aforementioned fields is hierarchical: AI represents the overarching goal of intelligent behavior; ML provides data-driven methods to approach this goal; DL offers specialized techniques within ML that leverage large datasets and computational resources to learn highly expressive representations; and, finally, GenAI is a specialized application of DL focused on content generation.

2.3. Natural Language Processing and Understanding

Natural Language Processing (NLP) is a multidisciplinary field at the intersection of computer science, linguistics, and artificial intelligence that focuses on enabling machines to process, analyze, and generate human language [43]. It comprises a broad range of tasks, including text preprocessing (e.g., tokenization, stemming, and lemmatization), syntactic parsing, part-of-speech tagging, named entity recognition (NER), and sentiment analysis. NLP leverages statistical models, rule-based systems, and deep learning architectures, particularly transformer-based models [293], to represent linguistic information at the lexical, syntactic, and semantic levels [43].

Natural Language Understanding (NLU) is a specialized subfield of NLP concerned with deriving semantic meaning and intent from human language [43]. While NLP may involve surface-level text processing, NLU aims to interpret linguistic input in a manner consistent with human comprehension, often requiring contextual reasoning, the disambiguation of polysemous terms, and the incorporation of world knowledge. Core NLU tasks include text classification and question answering, frequently relying on deep contextualized embeddings (see Section 2.3.1) and knowledge integration methods [43].

Recent advancements in NLU are closely tied to pre-trained LLMs and transfer learning (see Section 2.3.1), which allow systems to leverage vast amounts of unlabeled text to develop generalizable language representations. These representations are subsequently adapted to specific semantic tasks, often with minimal labeled data. Furthermore, prompting strategies (see Schulhoff et al. [255]) have emerged as an effective means of steering LLMs toward NLU objectives without explicit retraining.

In summary, NLP provides the computational and algorithmic foundation for language processing, while NLU focuses on semantic interpretation and reasoning. Together, they support applications such as conversational agents, information extraction systems, and intelligent search, forming a core component of modern human–computer interaction.

Table 2.2.: Overview of architecture types, the corresponding LLMs, and (proprietary) developers

Type	Developer	Language Model
Encoder	Google	BERT (2018) [54], ALBERT (2020) [169]
	Meta AI	RoBERTa (2019) [185]
	Microsoft	DeBERTa (2020) [98]
Decoder	Google	XLNet (2019) [320], LaMDA (2021) [284]
		PaLM, Gopher, Sparrow, Chinchilla, Minerva (2022)
		Bard (2023), Gemini 2.5 Deep Think[2] (2025)
	EleutherAI	GPT-J (2021), GPT-NeoX (2022), GPT-Neo (2023)
	Meta AI	OPT (2022), Galactica (2022) [281], LLaMA (2023) [287]
	Open AI	GPT-1 (2018), ..., CodeX (2021), InstructGPT (2022)
		ChatGPT (2022), GPT-4 (2023), GPT-5[3] (2025)
Encoder-Decoder	Google	T5 [229], Flan-T5 (2022), Flan-UL2 (2023)
	Meta AI	BART (2020) [179]

2.3.1. Large Language Models

Large Language Models (LLMs) constitute a class of neural network–based systems trained on extensive corpora to model and generate natural language. Their advances are attributable to efficient word representations in vector space (embeddings) and to the **Transformer architecture** (cf. Vaswani et al. [293], which introduced a self-attention mechanism capable of capturing long-range dependencies in text without the sequential bottlenecks inherent to recurrent neural networks (RNNs). In Transformers, input tokens are embedded into continuous vector spaces, processed in parallel through multi-head self-attention layers, and combined with position-wise feed-forward networks. The attention mechanism computes contextualized token representations by weighting all other tokens in a sequence according to learned relevance scores, enabling efficient scaling to billions of parameters. Based on the Transformer architecture, we can differ between three types of pre-trained language models, i.e., encoder, decoder, and encoder-decoder, as listed in Table 2.2.

Transfer learning plays a major role in the performance of LLMs. Instead of training from scratch for each task, LLMs are typically pre-trained on large-scale, domain-general text data using self-supervised objectives (e.g., masked language modeling or causal next-token prediction). This pre-training yields general linguistic and world knowledge encoded in the model parameters. Subsequently, models undergo fine-tuning or adaptation to specific tasks or domains, often requiring significantly smaller labeled datasets. Modern approaches also explore parameter-efficient adaptation methods (e.g., adapters, LoRA) to reduce computational costs while maintaining performance.

Prompting has emerged as a powerful paradigm for eliciting desired behavior from LLMs without explicit gradient-based fine-tuning. In this context, the model is conditioned

[2] `https://gemini.google/release-notes/` [Last accessed on 2025-08-13]
[3] `https://openai.com/gpt-5/` [Last accessed on 2025-08-13]

through carefully designed natural language instructions (prompts) or examples embedded in the input sequence. By providing a prompt to a GenAI model, it generates a corresponding response [255]. Researchers can set and vary the temperature (temp $\in [0, 1]$) of these models to influence the randomness and, thus, the creativity of the responses. Prompting techniques are a blueprint that specify how to structure and design a prompt [255]. For instance, this may include zero-shot prompting, where only the task description is given; few-shot prompting, where demonstrations are included; and chain-of-thought prompting, which guides the model to produce intermediate reasoning steps. In a recent survey, Schulhoff et al. [255] introduced a taxonomy of 58 text-based prompting techniques based on an analysis of 1,661 arXiv papers and six categories, namely *Zero-Shot*, *Few-Shot*, *Thought Generation (Chain-of-Thought (CoT))*, *Ensembling*, *Self-Criticism*, and *Decomposition*.

In summary, LLMs integrate the scalability of the Transformer architecture, the efficiency of transfer learning, and the flexibility of prompting to achieve state-of-the-art performance across a broad range of NLP tasks, such as text classification or question answering. Ongoing research aims to improve their factual accuracy, interpretability, and alignment with human values (including ethical considerations) while mitigating biases and hallucinations inherent in large-scale language modeling (cf. Section 2.3.2 and Section 10.1).

2.3.2. Retrieval-Augmented Generation

The term Retrieval-Augmented Generation (RAG) was first introduced by Lewis et al. [180] in the context of knowledge-intensive NLP tasks, such as question answering. The underlying concept has gained increasing attention as a promising and effective approach to address key limitations of LLMs, including hallucination, insufficient domain-specific and expert knowledge, and outdated information [213].

The core objective of RAG is to improve the outputs of an LLM w.r.t. factual accuracy by providing it with relevant information retrieved from an external knowledge source at the time of inference, instead of relying on the potentially outdated or imprecise knowledge internalized during pre-training. Thus, this paradigm is particularly advantageous in domains that demand high factual fidelity and rapid adaptation to emerging information, such as scientific research and technological innovation. RAG represents a hybrid approach that integrates information retrieval mechanisms with generative language models. The retrieval module identifies relevant information, e.g., documents, based on the input query, while the generation module synthesizes responses conditioned on both the query and the retrieved content. This integration enables LLMs to expand their effective knowledge scope, reduce hallucination rates, and produce more contextually grounded outputs. Despite its strengths, RAG systems exhibit inherent constraints. The factual accuracy of RAG-generated outputs depends on the quality of the retrieved information; irrelevant or misleading results can propagate errors, making retrieval precision and recall critical. Moreover, the retrieval process adds computational overhead and latency, which may impede real-time applications. Maintaining an up-to-date knowledge base index in dynamic domains incurs additional engineering and storage costs. Furthermore, RAG systems may inherit and amplify distributional biases present in the source corpus, posing challenges for fairness and robustness.

In summary, RAG offers a promising solution for improving the performance of LLMs in knowledge-intensive tasks. However, its effectiveness depends critically on the retrieval quality, system design, and domain-specific considerations.

2.4. From Data to Knowledge and Wisdom

The systematic handling of data, information, and knowledge within organizations and computational environments is frequently conceptualized through the (hierarchical) **Data, Information, Knowledge, Wisdom (DIKW)** pyramid. This model is widely cited based on the article *"From Data to Wisdom"* by Ackoff [2] from 1989. It serves as a foundational framework for understanding how raw data can be processed and contextualized into higher-order cognitive constructs. At its base, data represents raw symbols or measurements devoid of context. When data are organized and given relational meaning, they become information. Further contextualization and integration with experience or rules yield knowledge, which can inform decisions or actions. Finally, wisdom is often constructed as the judicious application of knowledge, incorporating ethical and temporal dimensions. As a result, each level of the pyramid represents a distinct type of cognitive processing and value addition. In conclusion, the DIKW pyramid became widely adopted in the 1990s and 2000s, particularly in the fields of knowledge management, library science, information systems, and information science [334, 240]. It illustrates the importance of transforming raw data into actionable knowledge and wisdom [69].

Within this hierarchy, **Knowledge Engineering (KE)** is the discipline concerned with the formalization and encoding of knowledge for use in AI and expert systems, a field that has been developed since the 1970s [254, 63]. It involves identifying, structuring, and validating knowledge to make it accessible for inference engines or decision support systems. Techniques such as ontological modeling, rule-based systems, and semantic networks are commonly employed in this domain. The overarching goal is to bridge tacit and explicit knowledge through representational structures that support computational reasoning [86].

Complementing this technical perspective is the field of **Knowledge Management (KM)**, which focuses on the creation, sharing, utilization, and retention of knowledge (within organizations) [51, 91]. KM aims to enhance organizational learning, innovation, and performance by aligning knowledge processes with strategic objectives. Approaches to KM are typically categorized into two paradigms: codification (explicit knowledge captured in repositories) and personalization (facilitation of tacit knowledge exchange through social interaction). Both paradigms are essential to achieving a sustainable competitive advantage in knowledge-intensive environments.

To operationalize KM practices, organizations deploy **Knowledge Management System (KMS)** [92] – integrated platforms that support the acquisition, storage, retrieval, and dissemination of knowledge. These systems often combine elements of document management, content management, collaboration tools, and AI techniques. Effective KMS are designed not merely as repositories but as dynamic ecosystems that enable knowledge flow, contextual relevance, and user engagement. Key success factors for KMS include organizational culture, user acceptance, and alignment with business processes.

In summary, these concepts form a comprehensive framework for understanding how knowledge is transformed, engineered, and managed, both within computational systems and organizational contexts. In our context, we regard the ORKG as a KMS that offers different views and management possibilities for the core feature of arranging and structuring scientific knowledge of research artifacts (cf. Section 2.6).

2.5. Knowledge Representation Approaches

In the following section, we present concepts and approaches to describe abstractions and knowledge representations of reality and to facilitate automated information processing.

2.5.1. Modeling Concepts

The concept of models is prevalent, occurring in technical and non-technical domains. Models provide a simplified representation of reality for a given purpose through abstraction. In various contexts, the term model is used differently. It can range from informal representations to mathematical formalizations [272]. According to the work of Stachowiak on general model theory [272], a model is specified by three main characteristics: representation, abstraction, and pragmatism [272, p. 131–133].

Representation. A model is a mapping or representation of the original it reflects. The original may be either natural or artificial, such as a physical object or an abstract concept.

Abstraction. A model captures only a subset of the properties of its original, deemed relevant by its creator. Consequently, models are abstract representations of originals driven by their pragmatism, defined as the third characteristic (i.e., pragmatism).

Pragmatism. A model is always designed and created for a specific purpose. Its relevance to the original depends on the intended users, the context of use, and the specific operations it is designed to support.

In SE, researchers and practitioners employ this definition and concept of models to abstract, represent, and manage complexity in software systems. For instance, in *Model-Driven Software Development (MDSD)* models are raised to first-class artifacts for software development. As a consequence, models are not solely used for documentation and communication, but also for generating executable code from them. Likewise, in *Software Product Line Engineering (SPLE)* models are used to express variability in a set of products and, thus, aim to promote the reuse of predefined software artifacts, i.e., core assets. In the *domain of AI* (Section 2.2), models are central to both the development and interpretation of intelligent behavior. Here, models serve as abstractions of the environment, decision-making policies, or internal representations of knowledge. For instance, symbolic models are employed in knowledge representation and reasoning systems, while statistical models are used in machine learning to capture patterns from data.

Detailed modeling techniques in these domains are described in the following. In this doctoral thesis, we use and adopt the concept and notion of models as structured representations to specify, document, communicate, and implement knowledge of research artifacts based on their corresponding metadata types.

2.5.2. Modeling Techniques for Knowledge Representations

In the following section, we provide an overview of different modeling techniques in terms of concepts, standards, and tools.

2.5.2.1. (Meta-)models and Modeling Formalisms

A metamodel is a model of a model (or a set of models) by specifying all types of model elements and relations between them. Therefore, each model can be considered as an instance of a metamodel (i.e., an instance-of relationship). As a metamodel is itself a model, one can define a metamodel of a metamodel. The Meta-Object Facility[4] (MOF) is a standard from the Object Management Group[5] (OMG; founded in 1989). The standard describes the relation between models and their metamodels as a four-layer hierarchy. In this hierarchy, Layer M0 represents real-world objects and constitutes the lowest level of abstraction. Layer M1 contains models, Layer M2 includes metamodels, and Layer M3 comprises a self-describing meta-metamodel, representing the highest level of abstraction. This layered architecture facilitates abstraction and reuse across modeling languages and tools. We use this kind of abstraction to define (digital) research artifacts, such as papers, software, and data, with their different (conceptual and technical) knowledge description approaches and corresponding knowledge bases (cf. Section 5.2.2).

2.5.2.2. Variability Modeling

Variability modeling is crucial for SPLE and constitutes a primary distinction from traditional software development practices [115]. Variability models provide a structured approach for representing the commonalities and variabilities within configurable systems and Software Product Lines (SPLs). They enable the explicit representation and specification of *variation points* (i.e., locations in the system where variability occurs) and their possible and associated *configurations* (i.e., selection of variants that together define a specific system instance), thereby facilitating the design, analysis, and automated derivation of product *variants* (i.e., specific options or realizations of variation points). This modeling capability enables the systematic reuse of software artifacts across a family of products, thereby enhancing scalability and maintainability. Another core concept in this context is a *feature*, defined as a user-visible characteristic or functionality that can vary across products. Several modeling approaches are employed to represent variability (i.e., different units of variability), such as feature modeling, decision modeling, orthogonal variability modeling (OVM), and UML extensions [115].

Feature Modeling The Feature-Oriented Domain Analysis (FODA) method, introduced by Kang et al. [128] (technical report, 1990), remains a widely adopted method for feature modeling. Feature models are typically visualized as hierarchical trees and provide a structured approach to capturing and managing the features of software products, facilitating the systematic development of customizable and high-quality software solutions.

[4] `https://www.omg.org/mof/` [Last accessed on 2025-08-14]
[5] `https://www.omg.org/` [Last accessed on 2025-08-14]

Tools and Languages A variety of tools and modeling languages have been developed to support feature modeling in software product line engineering. Among the most prominent is FeatureIDE[6], an integrated development environment introduced in 2005 [176], which provides comprehensive support for feature-oriented software development, including feature modeling, configuration, and code generation. Another notable advancement is the Universal Variability Language[7] (UVL) [26], which aims to standardize the representation of variability across tools and domains. UVL offers a flexible and extensible syntax for defining feature models, facilitating interoperability and tool integration. These tools and languages contribute significantly to the formalization, automation, and scalability of variability management in complex software systems.

2.5.2.3. Knowledge Graphs

A knowledge graph (KG) is a structured representation of information in the form of a graph, where nodes represent entities and edges represent relationships between those entities. KGs have been used in the literature since at least 1972 [103]. Each node and edge is typically annotated with semantic metadata, enabling machines to interpret and reason over the graph's content. Knowledge graphs are grounded in principles from graph theory, semantic web technologies, and ontology engineering. They are often implemented using `RDF` (Resource Description Framework) triples of the form $\langle$subject, predicate, object$\rangle$, which encode factual statements [103]. Knowledge graphs support semantic interoperability, information retrieval, and automated reasoning by integrating heterogeneous data sources into a machine-readable format. They are widely used in domains such as NLP, question answering, recommendation systems, and scientific knowledge management.

Research Knowledge Graphs A Research Knowledge Graph (RKG), also referred to as a scientific, academic, or scholarly knowledge graph, is a specialized form of a KG designed to represent scholarly entities and their relationships in a semantic, machine-actionable, and graph-based format. RKGs utilize formal classification and data schemas to encode entities such as publications, authors, institutions, and related research outputs. This structure enables complex, multi-hop queries and supports the retrieval of contextual and provenance information, which are essential for transparency, reproducibility, and accountability in scientific workflows. RKGs play a critical role in implementing the FAIR principles by integrating persistent identifiers (e.g., DOIs, ORCIDs) and semantic ontologies to ensure interoperability and machine interpretability. They also enhance metadata structuring and alignment, improving discoverability and reuse across platforms and disciplines. In summary, RKGs serve several functionalities in the scientific context, i.e., a) Data integration; b) Data FAIRification; c) Metadata cataloguing; d) Data verification, enrichment, and extension; e) Machine-actionability and AI-readiness [237]. Currently, we can differ between two types of RKGs, which offer a representation of research artifacts (i.e., papers and supplementary material):

1. **Bibliographic metadata** of research artifacts (titles, authors, venues, etc.)

[6] `https://featureide.github.io/` [Last accessed on 2025-09-27]
[7] `https://universal-variability-language.github.io/` [Last accessed on 2025-09-27]

Examples: dblp KG[8] [1], OpenAIRE[9], Semantic Scholar Academic Graph (S2AG) [297], and OpenAlex [226]

2. **Knowledge-based metadata** of research artifacts (focus on content-related data)

   Examples: Computer Science Knowledge Graph[10] [53, 52], Open Research Knowledge Graph (ORKG) [13, 117, 109] (see Section 2.6)

Both types of RKGs capture descriptive metadata (cf. Definition 2.1.2).

2.6. The Open Research Knowledge Graph

The ORKG is an instance of an RKG and functions as a KMS (cf. Section 2.4) for research artifacts. Its primary objective is to semantically organize scientific data and knowledge across diverse research fields[11]. Consequently, the ORKG provides a digital infrastructure, dedicated to support the generation, curation, dissemination, and reuse of semantic scholarly knowledge following the FAIR principles [117]. The underlying graph stores scientific articles and their statements as contribution entities. Registered users are permitted to add new entries, while the quality of contributions is monitored by administrators. As of now, the graph contains 731 research fields, 37,562 papers, 1,547 ORKG comparisons, and 826 ORKG templates[12]. In the following, we introduce the data model and core content types of the ORKG, which are described and elaborated on in further detail in [109].

Data Model The ORKG employs a knowledge graph-based data model. The term KG (Section 2.5.2.3) originates from the Semantic Web domain, which extends the capabilities of the World Wide Web (WWW) by enabling the interlinking of data rather than documents. The ORKG adopts this paradigm by semantically describing research contributions through structured, interrelated data elements. Importantly, while the ORKG follows Semantic Web principles, interaction with the system does not require users to possess technical knowledge of the domain. The ORKG User Interface is designed to be accessible to users without prior experience in Semantic Web technologies.

At the core of the Semantic Web is the `RDF`[13], a specification used by the ORKG to represent information. In `RDF`, knowledge is modeled as triples, each consisting of a subject, predicate, and object. These triples, also referred to as statements (cf. Table 2.3), are inspired by linguistic constructs, where the subject and object denote entities or values, and the predicate describes the relationship between them. The subject and object positions can represent resources, classes, or properties. The predicate position exclusively represents properties. The object position may also include literals, which are atomic values such as strings that are not further decomposed or linked to other resources.

[8] `https://github.com/dblp/kg` [Last accessed on 2025-08-14]
[9] `https://graph.openaire.eu/` [Last accessed on 2025-08-14]
[10] `https://w3id.org/cskg` [Last accessed on 2025-08-14]
[11] `https://orkg.org/fields` [Last accessed on 2025-08-13]
[12] `https://orkg.org/stats` [Last accessed on 2025-08-13]
[13] `https://www.w3.org/RDF/` [Last accessed on 2025-09-13]

To facilitate reference and reuse, the ORKG automatically assigns unique identifiers (IDs) to all entities, including resources, properties, and classes. Additionally, by associating a class with a resource, the resource is defined as an instance of that class. Although class assignments are optional in the ORKG, their use enhances semantic clarity and supports one of the ORKG's primary goals: the structured and meaningful organization of scholarly knowledge. Figure 2.2 depicts a basic ORKG schema for papers.

Table 2.3.: Types of statements in the ORKG

Subject	Predicate	Object
{ Resource, Property, Class }	Property	{ Resource, Property, Class, Literal }

Within the ORKG system, frequently used concepts are referred to as **Content Types**. These Content Types are represented by dedicated pages in the ORKG user interface and conform to a predefined data model, as previously described. This model enables users to describe scholarly knowledge in a structured and semantically meaningful manner. While the ORKG supports flexible knowledge representation, Content Types play a critical role in promoting standardization and interoperability. By enforcing consistent structural patterns, Content Types enhance the **machine-actionability** of the data, facilitating automated processing, comparison, and integration of research contributions. In the following, we provide a detailed overview of the most prominent Content Types of ORKG.

Research Fields Research contributions in ORKG, including papers, are systematically categorized under specific Research Fields[14] (Figure 2.2). This taxonomy enhances the semantic organization of knowledge within the graph and facilitates content retrieval based on disciplinary domains. The **ORKG Research Field taxonomy**[15] is adapted from the taxonomies developed by the National Academies of Sciences, Engineering, and Medicine, the German Research Foundation (DFG), and the ArXiv Category Taxonomy. The ORKG taxonomy is curated by the ORKG team to ensure consistency, accuracy, and broad disciplinary coverage.

Building upon this hierarchical taxonomy structure and as a research field's representation in the ORKG increases, researchers can run an **Observatory**[16]. This is a domain-specific entry point that aggregates, analyzes, and visualizes research contributions within a given *Research Field*. Observatories offer curated views of the knowledge graph tailored to particular scientific domains, enabling researchers and other stakeholders to track the evolution of scientific knowledge, identify emerging trends, and perform structured comparisons of research contributions (see ORKG Comparison).

Each Observatory is automatically generated and continuously updated, leveraging the structured nature of ORKG content and its alignment with standardized Content Types and Research Fields. Through these observatories, the ORKG not only facilitates knowledge

[14] `https://orkg.org/fields` [Last accessed on 2025-08-13]

[15] `https://orkg.org/help-center/article/20/ORKG_Research_fields_taxonomy` [Last accessed on 2025-09-13]

[16] `https://orkg.org/observatories` [Last accessed on 2025-09-13]

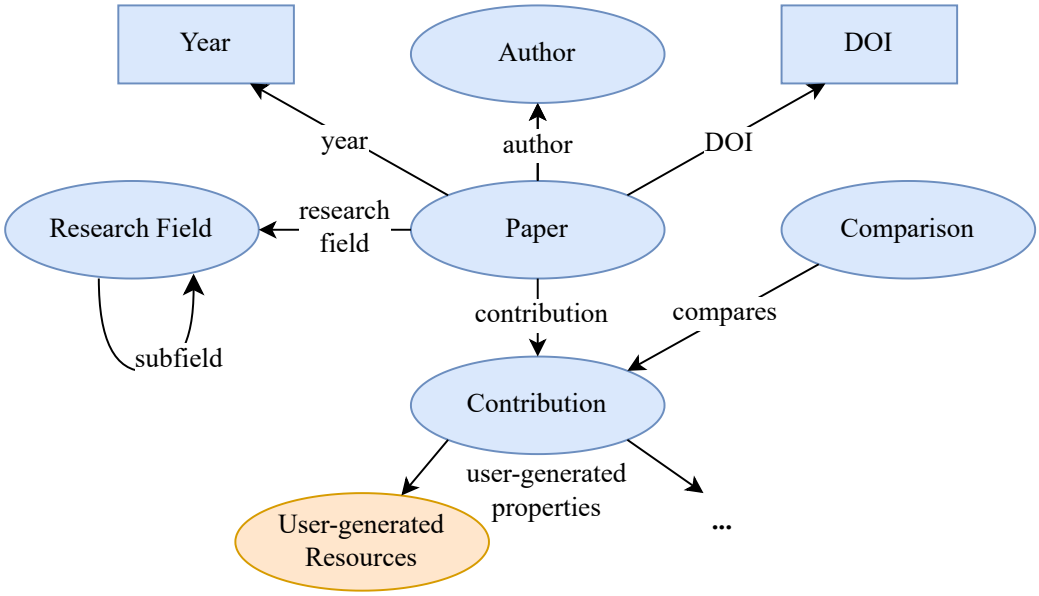

Figure 2.2.: Basic ORKG schema for papers. Notation: Blue shapes are resources and predicates of the ORKG schema. Yellow shapes are user-generated resources and predicates. IDs and classes of resources and predicates are omitted for brevity [12]. (Figure adapted from Auer et al. [12])

discovery but also promotes transparency and reproducibility in scientific research by exposing the underlying data structures and semantics.

Contribution Templates An ORKG contribution template[17] is a predefined semantic structure designed to standardize the representation of research contributions. These templates serve as a foundational mechanism for ensuring that contributions addressing similar research problems conform to a consistent data model, thereby enhancing comparability, interoperability, and machine-readability. A contribution template functions as a schema or skeleton for describing the key elements of a research contribution. It defines: (i) the **target class** of contributions (e.g., algorithm, experiment, material), (ii) **properties** that must be specified (e.g., methodology, results, dataset used), and (iii) the **cardinality and data types** for each property (e.g., exactly one, optional, string, date, or number). This structured approach allows users to input data in a "fill-in-the-blank" format, reducing ambiguity and facilitating semantic alignment across contributions. Templates are particularly useful for: (i) **creating comparisons** between studies, (ii) automating **data extraction and integration**, and (iii) improving the **quality and consistency of metadata**. Templates can be created by domain experts manually via the ORKG interface (see Figure 2.3). Once defined, they can be reused across multiple contributions and papers, enabling scalable and consistent knowledge modeling. Figure 2.3 depicts the ORKG template system.

[17] `https://orkg.org/about/19/Templates` [Last accessed on 2025-09-13]

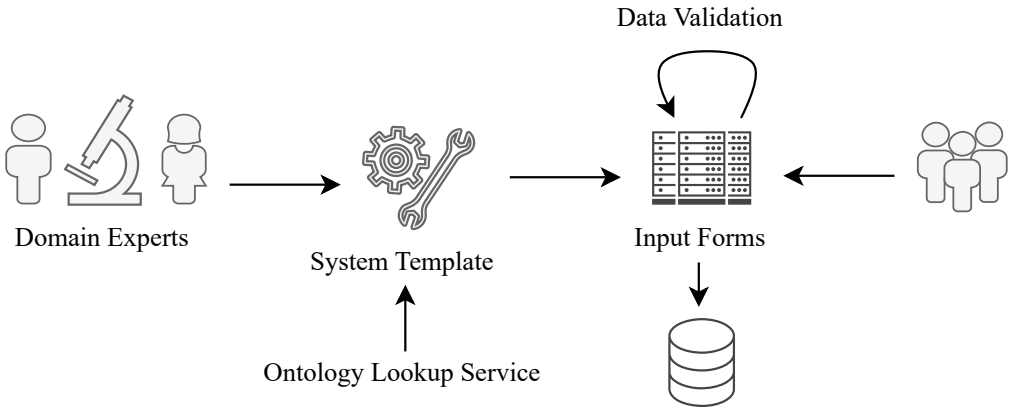

Figure 2.3.: ORKG template system according to Ilangovan et al. [109]

ORKG Comparison An ORKG Comparison[18] is a structured, semantic representation of (research) contributions that enables the systematic comparison of scientific knowledge across multiple publications. They serve as a core content type within ORKG and provide an overview of the state-of-the-art for a particular research question. Each comparison aggregates contributions to a specific research question and organizes them in a tabular format. The rows in the table represent individual studies or papers, and the columns denote comparable properties (e.g., methodology, results, datasets used, evaluation metrics). The comparison is enriched with semantic annotations, enabling advanced filtering, sorting, and visualization. These features support users in identifying patterns, trends, and gaps in the literature. ORKG Comparisons are built upon semantic web technologies and knowledge graph principles, which allow for: (i) interoperability with other data sources and ontologies, (ii) machine-actionable metadata, facilitating automated reasoning and data integration, and (iii) FAIR data practices (cf. Section 2.1.3). They are particularly valuable for meta-analyses, systematic reviews, and evidence synthesis, where the structured comparison of multiple studies is essential. The ORKG platform supports the creation of comparisons, e.g., through manual input (via a guided interface), structured templates (for common comparison types), and `CSV` import (allowing the bulk addition of contributions and properties). Comparisons can be exported in various formats (e.g., `CSV`, LaTeX, `RDF`, `PDF`) and formally published with persistent identifiers (DOIs) for citation and reuse.

2.7. Sustainable Software Development and Green AI

The **Karlskrona Manifesto for Sustainability Design**[19] was originally drafted in 2014 at the Third International Workshop on Requirements Engineering for Sustainable Systems[20] (RE4SuSy) held in Karlskrona, Sweden, co-located with the 22nd IEEE International

[18] `https://orkg.org/about/15/Comparisons` [2025-09-13]
[19] `https://sustainabilitydesign.org/` [Last accessed on 2025-04-22]
[20] `https://ceur-ws.org/Vol-1216/` [Last accessed on 2025-04-22]

Requirements Engineering Conference (RE 2014) [23]. Until now, this manifesto serves as a foundational document, aiming to establish a common ground for integrating sustainability principles into software design and development. Therefore, it addresses the urgent need for a more sustainable world by *highlighting misperceptions and misconceptions* (e.g., framing sustainability as only protecting the environment while ignoring, e.g., social and economic concerns) and providing a fundamental *set of guiding principles and commitments* for sustainability design within the software community and beyond. As a consequence, it initiates the *dialogue among different stakeholders* on sustainability in software design in a holistic way, while aiming to create a common ground and unify the software engineering community around sustainability goals w.r.t. individual, social, economic, environmental, and technical aspects [22].

With the rapid adoption and reported progress of AI-based systems, the field of Red AI has continued to expand. Red AI refers to traditional AI research that focuses primarily on improving accuracy (or similar measures) through massive computational power. In 2019, Schwartz et al. [256, 257] were the first to advocate for increased research in Green AI, an alternative approach that aims to make AI development more environmentally friendly and inclusive. Thus, Green AI refers to the integration of environmental sustainability principles into AI development and deployment by addressing the significant carbon footprints associated with training large AI models, particularly in natural language processing and deep learning, while maintaining their performance. This also impacts and increases inclusivity by decreasing the barriers to participation in NLP research.

Comparison The *Karlskrona Manifesto* and *Green AI* both address sustainability aspects but focus on different domains and approaches. The Karlskrona Manifesto emphasizes the need for sustainable design principles within SE, advocating for a collective understanding of sustainability's impact on software systems. In contrast, Green AI concentrates on enhancing the efficiency of artificial intelligence research to reduce its carbon footprint alongside traditional performance measures. This distinction highlights the varied approaches to achieve sustainability across different technological fields.

2.8. Evaluation Preliminaries

Finally, we introduce the basic building blocks required for our evaluations in Part II and Part III. First, we present the underlying approach to operationalize our method designs, i.e., the GQM approach. Second, we explain all relevant properties (i.e., the characteristics of evaluation intention that can be formulated as a validation question), common metrics, and evaluation concepts (e.g., sustainability in terms of energy consumption) for this thesis that build, in general, the basic layers of every GQM evaluation plan.

2.8.1. Goal-Question-Metric Approach

The GQM approach was originally developed during the 1980s by Basili et al. [20] to provide a structured mechanism for feedback and to assess software development results. About a decade later, the concept was further extended and specified by Basili et al. [19].

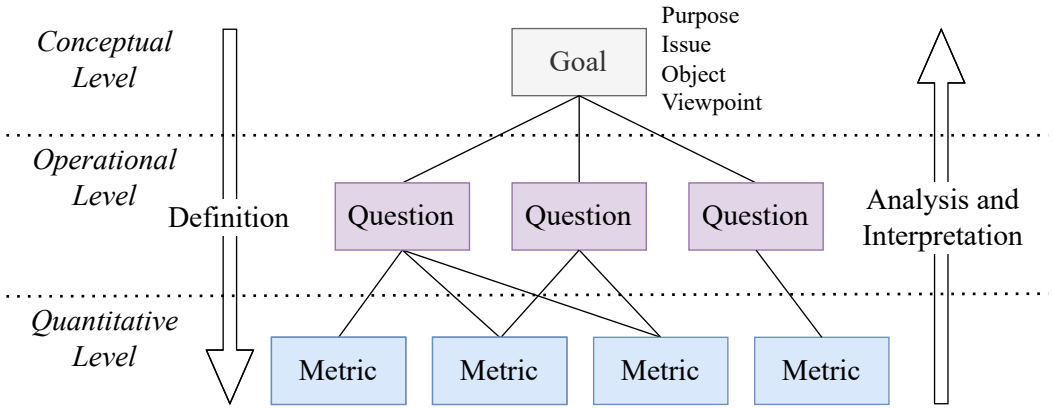

Figure 2.4.: The Goal Question Metric (GQM) approach (Figure extended and adapted from Basili et al. [19])

The approach employs a hierarchical structure by linking goals, questions, and metrics with the general idea of define measurements in a top-down fashion. Therefore, **Goals** are defined first, and they can be defined for a variety of objects of measurement ranging from products and processes to resources. As such, the GQM approach can also be applied to research objects, especially for SE research. Based on the underlying reason, the goals shape the overall direction and concept of the evaluation. As a result, a goal defines a *purpose* of measurement, an *issue* to be measured, an *object* (or process) to be measured, and a *viewpoint*. **Questions** are then used to characterize the object of measurement, based on certain quality criteria. They define the evaluation on an operational level and break down the goal into its components. Finally, the questions need to be answered quantitatively with the help of **Metrics**. The data can be either objective or subjective. Objective data only depends on the (research) object that is measured, not on a certain viewpoint. Data is subjective if it depends on the viewpoint from which it is collected and on the object. Metrics are not unique in the sense that the same metrics can be used to answer different questions under the same goal.

The advantages of such a top-down fashion over a bottom-up one are the streamlined and focused way to select just the necessary, specific set of metrics out of all possible observable metrics. Therefore, the approach can help to target the exact set of facets that should be covered with an evaluation. Figure 2.4 depicts the general principle of the GQM approach w.r.t. the conceptual (goal), operational (question), and quantitative (metrics) level structure as well as the faceted goal specification.

We use and enhance this approach in our evaluation approach for classification and data schemas (cf. Section 5.3) to define and operationalize the corresponding research method design in our evaluations presented in Part II and Part III.

2.8.2. Properties for Language Evaluation

Guizzardi et al. [88] (2005) propose an ontology-based framework for evaluating and (re-) designing modeling languages, with a particular focus on two key properties: domain appropriateness and comprehensibility appropriateness. These properties serve as criteria

for assessing how well a modeling language supports the accurate representation of real-world phenomena and facilitates user understanding and communication.

Following the work of Gurr [89], the authors define *lucidity*, *soundness*, *laconicity*, and *completeness* as further underlying and formal criteria for their language evaluation framework (see Figure 2.5).

Lucidity refers to the clarity and unambiguity of a model, i.e., one has to ensure that each model element has a well-defined and distinct meaning, avoiding confusion or semantic overlap (see Figure 2.5 (a)).

Soundness refers to the validity of all elements in a model, i.e., one has to prevent the use of (classification) constructs that represent impossible or incoherent states, avoiding logical inconsistencies and misrepresentations (see Figure 2.5 (b)).

Laconicity refers to the essential (classification) constructs needed (i.e., the minimal number of constructs) to represent real-world phenomena without duplicating or fragmenting their representation across multiple elements, i.e., one has to avoid redundancy and over-specification in models (see Figure 2.5 (c)).

Completeness refers to the guarantee that all relevant aspects are represented in the model, i.e., one has to avoid omissions that could lead to an incomplete or misleading understanding (see Figure 2.5 (d)).

These properties are not merely stylistic or aesthetic guidelines; they are formal ontological constraints that can be encoded in the metamodel of a language (e.g., via OCL rules) to enforce quality and consistency across models. Together, they support the creation of models that are semantically rigorous, cognitively accessible, and pragmatically efficient. This work emphasizes the transferability of evaluation concepts between and across different knowledge description approaches.

We use the work of Guizzardi et al. [88] to derive metrics for classification and data schemas to evaluate their structure's suitability, embedded in a GQM plan, supporting and constructing our evaluation approach over multiple research steps (cf. Section 5.3).

2.8.3. Inter-Annotator Agreement

The Inter-Annotator Agreement (IAA), also known as inter-annotator reliability (IAR), specifies the degree of consensus or similarity between annotations made by multiple annotators who independently assign labels to the same dataset for a classification task. It serves as an indicator of the **reliability** of the annotation process and, by extension, the validity of subsequent analyses that depend on these annotations. A high IAA suggests that the annotation schema is well-defined, unambiguous, and likely reproducible across different annotators; low IAA may indicate insufficient clarity in definitions, inadequate annotator training, or inherent ambiguity in the task. In empirical SE research, NLP, and other empirical sciences, the IAA is particularly important when manual coding or qualitative classification is used, such as tagging software architecture documentation, identifying design decisions, or classifying research papers.

IAA can be measured using a variety of metrics, which differ in their assumptions, sensitivity to chance agreement, and applicability to categorical, ordinal, or continuous data. Each of them has specific benefits and drawbacks. Commonly used metrics are Percent

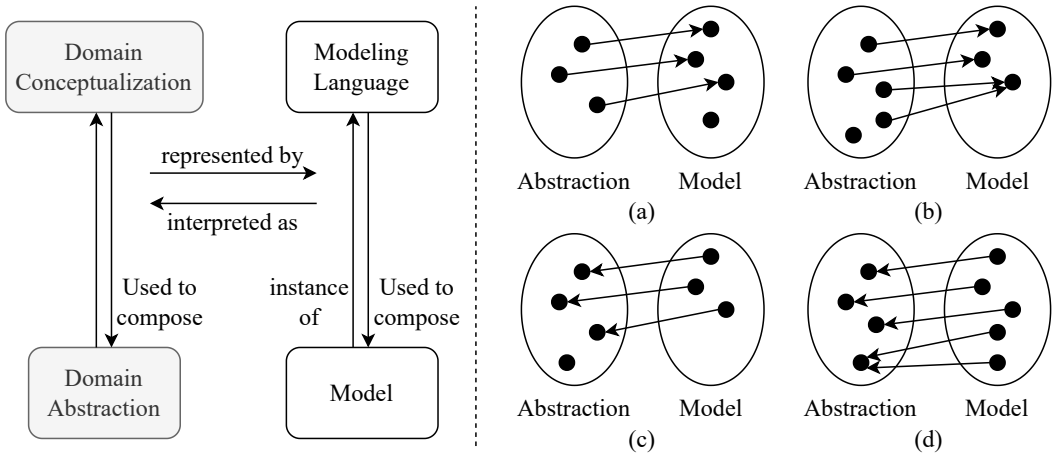

Figure 2.5.: Relation between conceptualization, abstraction, modeling language, and model according to Guizzardi et al. [88].
Examples of (a) *Lucid* and (b) *Sound* representational mappings from (domain) Abstraction to Model; Examples of (c) *Laconic* and (d) *Complete* interpretation mappings from Model to (domain) Abstraction.

Agreement [106], Bennett's S [27], Cohen's κ [45], Fleiss' κ [68], Krippendorff's α [160], and Scott's π [258]. We describe these metrics in the following:

- Percent Agreement (PA) calculates the proportion of data items for which all annotators assign the same label. This metric permits calculating the agreement between two or more annotators.

$$\text{PA} = \frac{\text{Number of agreements}}{\text{Total number of data items}} \tag{2.1}$$

  The PA is very intuitive and simple to compute. However, it overestimates agreement by ignoring chance.

- Bennett's S is easy to interpret and can be applied to nominal data. However, the metric is less robust than kappa and assumes equal probability of all categories. This metric permits calculating the agreement between 2 annotators.

- Cohen's κ works best with large sample sizes. This metric permits calculating the agreement between 2 annotators.

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \tag{2.2}$$

- Fleiss' κ extends Cohen's κ to multiple annotators. It works best with large sample sizes, as it is sensitive to prevalence and category imbalance.

- Krippendorff's α handles any data type (nominal, ordinal, interval, or ratio). It supports missing data and is flexible for unequal ratings. This metric permits calculating

Table 2.4.: Relation between the Kappa statistic and the strength of agreement (assigned labels)

κ Value	Landis & Koch (1977) [170]
0.81 - 1.00	Almost Perfect
0.61 - 0.80	Substantial
0.41 - 0.60	Moderate
0.21 - 0.40	Fair
0.20 - 0.00	Slight
< 0.00	Poor

the agreement between 2 or more annotators. In summary, it is a good choice as it is very flexible and can deal with things such as incomplete data, varying sample sizes and various categories. However, it is more complex to compute.

- Scott's π is a widely accepted metric and works for nominal data. This metric permits calculating the agreement between 2 annotators.

The results of these metrics need to be interpreted. An acceptable level of agreement depends on the domain and application. According to Krippendorff [161], it is customary to require a value above 0.80. Similarly, Landis et al. [170] define that a value between 0.41 and 0.60 shows a moderate agreement, a value between 0.61 and 0.80 shows a substantial agreement, and a value above 0.80 shows almost perfect agreement. Table 2.4 provides an overview of the calculated values and categorization of strength of agreement.

2.8.4. Performance Metrics in Machine Learning

Selecting suitable metrics for a classification task is crucial. Researchers have to argue about an appropriate selection for a given scientific and experimental problem. To evaluate the classification task, we need a gold standard dataset (labeled by human experts) to assess the applicability of a knowledge description approach (cf. Section 5.3.2). We introduce evaluation metrics for our proposed evaluation approach in Section 5.3 and the experiments in Chapter 12. For classification tasks, we differ between single-class, multi-class, and multi-label problems (cf. Section 5.2.3). In the following, we present performance metrics / similarity scores and formulas expressed for single-class classification problems. The scores of the following metrics lie in the closed interval [0,1].

Confusion Matrix A confusion matrix (cf. Table 2.5) is a fundamental tool for evaluating the performance of classification models. It provides a tabular representation of the actual versus predicted classifications, allowing for the computation of various performance metrics and similarity scores [158].

Performance Metrics (Prediction) Based on the confusion matrix, researchers calculate various performance metrics, which we introduce in the following.

Table 2.5.: Confusion matrix for binary classification

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- **Accuracy** is the proportion of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

- **Precision** is the proportion of true positives among all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.4}$$

- **Recall (Sensitivity)** is the proportion of true positives among all actual positives:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.5}$$

- **Specificity** is the ratio between all true negative labeled objects and all in reality negative objects:

$$\text{Specificity (Spec)} = \frac{TN}{TN + FP} \tag{2.6}$$

- F_1 **Score** is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.7}$$

Similarity Scores In classification tasks, similarity scores are used to evaluate how closely a model's predictions match the true labels. These metrics are especially important for multi-label classification problems, where the nature of prediction and error measurement differs. Two commonly used metrics are Jaccard Score and Hamming Loss.

- **Jaccard Score** measures the overlap between predicted and actual labels. It is particularly useful in multi-label classification, where it quantifies the similarity between sets of labels. In single-label classification tasks, it simplifies to a ratio of correctly predicted instances over the union of predicted and actual instances. The Jaccard Score can also be calculated using the averaging strategies macro, weighted, micro, and sample based [269].

$$\text{Jaccard Score (JS)} = \frac{TP}{TP + FP + FN} \tag{2.8}$$

  A score of 1 means perfect overlap between predicted and actual labels; 0 means no overlap. Thus, higher values are better.

- **Hamming Loss** calculates the fraction of labels that are incorrectly predicted. In multi-label classification, it accounts for errors across all labels per instance. In single-label classification, it reduces to the proportion of incorrect predictions across all samples [268].

$$\text{Hamming Loss (HL)} = \frac{FP + FN}{\text{all labels}} \tag{2.9}$$

  A score of 0 means all labels are correctly predicted; higher values indicate more errors. Thus, lower values are better.

These metrics provide complementary insights: Jaccard Score emphasizes correct label matches, while Hamming Loss highlights prediction errors. Together, they help assess model performance from both accuracy and error perspectives.

2.8.5. Sustainability Aspects

A central concern in the field of Green AI (Section 2.7) and a different aspect, besides performance measures (Section 2.8.4), is the evaluation of sustainability, with particular emphasis on the environmental footprint of AI systems. The computational processes and costs involved in training and inference, especially those utilizing LLMs on Graphics Processing Units (GPUs), are associated with substantial electricity consumption, energy usage, and carbon emissions. Energy usage in these contexts is typically quantified in kilowatt-hours (kWh) or megajoules (MJ), with the conversion rate defined by the International System of Units[21] (SI): $1kWh = 3.6MJ$.

Electricity consumption contributes to environmental degradation when the energy is sourced from non-renewable resources such as fossil fuels (e.g., coal, oil, natural gas) or nuclear power. This environmental impact is commonly expressed as the carbon footprint, which encompasses the emission of greenhouse gases, including carbon dioxide (CO_2), methane (CH_4), nitrous oxide (N_2O), and fluorinated gases. These emissions are standardized and reported in terms of carbon dioxide equivalents (kg CO_2eq) [186].

To quantify carbon emissions, researchers must first estimate or measure the total energy consumption of their AI systems. This value is then converted into CO_2 equivalents by applying a location-specific conversion factor that reflects the local energy mix, i.e., the proportion of electricity generated from renewable sources (e.g., solar, wind) versus non-renewable sources (e.g., coal, gas, nuclear) [186].

Accurate measurement and estimation of both energy consumption and carbon emissions require a stable and controlled experimental setup. Consistency in hardware and location is essential, as variations in machine efficiency and regional energy profiles can significantly affect the results. For instance, identical workloads executed on different machines or in different geographical locations may yield divergent energy and carbon footprints due to differences in hardware performance and electricity sourcing.

Transparent documentation and reporting of these metrics are critical for a comprehensive assessment of an AI system's environmental impact. Such practices not only facilitate reproducibility and comparability across studies but also contribute to the broader goal of

[21] `https://www.iec.ch/si` [Last accessed on 2025-07-29]

reducing the ecological footprint of AI technologies. By integrating sustainability metrics into the evaluation and development lifecycle, researchers can promote the advancement of more environmentally responsible and energy-efficient AI systems.

In this doctoral thesis, we derive combined metrics to consider performance aspects and environmental impact to enable trade-off decisions (cf. Section 5.3.2.2) and provide best practices for the experiment setting and conduction (cf. Section 10.2).

2.8.6. Metrics for Retrieval-Augmented Generation Systems

The following sections present a set of performance metrics that are relevant to systematically assess and evaluate RAG systems. To this end, we introduce and explain both metrics to assess the quality of the (i) retrieval and (ii) generation components of RAG systems that we use for our experiments in Chapter 13 to evaluate the graph-based RAG approaches. For calculating the metrics, we use Retrieval Augmented Generation Assessment (RAGAs)[22] [59], a framework for reference-free evaluation of RAG pipelines. Furthermore, we use the established definitions and formulas of Yu et al. [325], Alinejad et al. [6], and Ibrahim et al. [107]. The following metrics are *single-turn metrics*, i.e., they evaluate the performance of the AI application based on a single turn of interaction between the user input (question) and the AI system (answer).

2.8.6.1. Metrics for the Retrieval Component

For the evaluation of the retrieval quality, we use exclusively *Non-LLM-based* metrics (i.e., deterministic metrics, relying on traditional methods to evaluate the performance of the retrieval). We differ between rank-agnostic metrics (non-rank-aware metrics) and rank-aware metrics. The scores of the following metrics lie in the closed interval [0,1], where higher values indicate better performance, unless stated otherwise.

Rank-agnostic Metrics These types of metrics assess the quality of retrieval without taking into account the position of an item within the list of retrieved contexts. For the corresponding equations, please refer to Section 2.8.4.

- **Accuracy** provides an assessment of the ratio of correctly retrieved contexts compared to the total number of retrieved contexts.
- **Precision** quantifies the fraction of retrieved documents that are relevant, indicating the retrieval system's ability to avoid false positives.
- **Recall** quantifies the fraction of relevant documents that are successfully retrieved, capturing the system's completeness in finding information.
- F_1 **Score** calculates the harmonic mean of precision and recall, providing a balanced measure that penalizes systems performing poorly on either metric.

[22] `https://docs.ragas.io/en/latest/concepts/metrics/` [Last accessed on 2025-04-22]

Rank-aware Metrics These types of metrics additionally evaluate the order in which relevant items are presented. Typically, the higher a relevant item is placed in the list, the better the score.

- **Hits@k** calculates the fraction of correctly retrieved contexts that appear in the top-k total of retrieved contexts.

$$\text{Hits@k} = \frac{H_k}{N_{query}} \tag{2.10}$$

  where:

  - H_k is the number of times a relevant context entry is in the top-k
  - N_{query} is the total number of retrieved context

- **Mean Reciprocal Rank (MRR)** measures the average of the inverse ranks of the first relevant context retrieved by the system. This means that the metric focuses on how high the retriever ranks the first relevant context.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{2.11}$$

  where:

  - |Q| represents the total number of queries
  - $rank_i$ denotes the rank position of the first relevant document for the i-th query

- **Mean Average Precision (MAP)** is the mean of the average precision scores for each query.

$$MAP = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{\text{relevant documents}_q} \tag{2.12}$$

  where:

  - $P(k)$ is the precision at cutoff k in the list
  - $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant document, 0 otherwise
  - n is the total number of retrieved documents

- **Exact Match (EM)** is the proportion of predictions (responses / retrieved contexts) that match the reference exactly. This metric is used in tasks requiring precise matching. The metric returns 1 if the response is an exact match with the reference, and 0 otherwise.

$$\text{EM} = \frac{PEM}{N_{pred}} \tag{2.13}$$

  where:

- PEM is the proportion of exact matches
- N_{pred} is the total number of predictions

In the context below, true positives (TP) refer to the documents that are relevant to the query and are accurately identified and retrieved by the system. True negatives (TN) are documents that are not relevant to the query and are correctly excluded from the retrieval results. False positives (FP) are the irrelevant documents that the system erroneously retrieves. False negatives (FN) are the relevant documents that the system fails to retrieve.

2.8.6.2. Metrics for the Generation Component

For the evaluation of the answer generation quality, we use traditional text generation metrics and LLM-as-a-Judge methods (i.e., dynamic evaluation by prompting a model). The scores of the following metrics lie in the closed interval [0,1], where higher values indicate better performance.

Traditional Text Generation Metrics These types of metrics refer to algorithmic and deterministic evaluation. These metrics assess the quality of generated text by comparing it to one or more reference texts (golden answer / ground truth) using predefined mathematical formulations. While these metrics are computationally efficient and reproducible, they may exhibit limitations in capturing nuances of meaning, factual correctness, and stylistic appropriateness, especially in cases where valid outputs diverge lexically from the reference texts. They can be broadly categorized into **semantic similarity metrics** (BERTScore, Semantic Similarity) and **lexical overlap metrics** (BLEU, ROUGE, String Similarity).

- **BERTScore**[23] [331] uses pre-trained Transformer embeddings to compute semantic similarity rather than lexical overlap and is particularly suitable for scientific domains with paraphrasing and terminology variations.
- **Semantic Similarity** calculates the cosine similarity between the generated answer and the golden answer (ground truth answer). To calculate this metric, the following steps have to be proceed:
  1. Vectorize the ground truth answer using the specified embedding model (ground truth embedding v_1).
  2. Vectorize the generated answer using the same embedding model (generated answer embedding v_2).
  3. Compute the *cosine similarity* between the two vectors (i.e., $cosine(v_1, v_2)$).
- **Bilingual Evaluation Understudy (BLEU)** [214] measures n-gram overlap between generated and reference answers, originally for machine translation but widely adopted for text generation.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.14}$$

[23] `https://github.com/Tiiiger/bert_score` [Last accessed on 2025-04-22]

- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE)** [184] calculates n-gram, word sequence, and longest common subsequence overlap between generated and reference texts, capturing lexical precision and content coverage essential for factual accuracy.
- **String Similarity** calculates the *Levenshtein distance* between the generated answer and the golden answer (ground truth answer) without relying on LLMs.

LLM-as-a-Judge Complementing these traditional text generation metrics, we employ LLM-as-a-judge evaluation methods and metrics that leverage the semantic understanding capabilities of LLMs to assess answer quality in ways that capture nuances beyond lexical similarity.

- **Faithfulness** measures the factual consistency of generated answers against retrieved contexts, ensuring that all claims in the answer can be logically inferred from the provided evidence. This is crucial for scientific QA where accuracy and verifiability are paramount.

$$\text{Faithfulness} = \frac{C_i}{C_n} \tag{2.15}$$

  where:
  - C_i is the number of claims in the response supported by the retrieved context
  - C_n is the total number of claims in the response

- **Answer Relevancy** evaluates how well the generated answer addresses the specific question asked, penalizing responses that are incomplete, overly verbose, or contain irrelevant information.

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^{N} \text{cosine similarity}(E_{g_i}, E_o) \tag{2.16}$$

  where:
  - N denotes the number of generated questions
  - E_{g_i} is the embedding of the i-th generated question
  - E_o is the embedding of the actual (user) question

- **Factual Correctness** assesses the alignment between generated answers and ground truth references, providing an evaluation of answer accuracy. To evaluate the degree of alignment between the generated response and the reference text, the metric employs an LLM to decompose both texts into discrete factual claims. Subsequently, Natural Language Inference (NLI) is applied to assess the extent of factual correspondence between the two claim sets. The formula for calculating True Positive (TP), False Positive (FP), and False Negative (FN) is as follows:

TP = Number of claims in the response that are present in the reference
FP = Number of claims in the response that are not present in the reference
FN = Number of claims in the reference that are not present in the response

Using these values, the Precision, Recall, and F_1 scores are subsequently calculated in accordance with the definitions provided above.

- **Instruction Following** evaluates whether generated answers comply with explicit formatting requirements in questions (e.g., ordering, aggregation, specific output structures). This metric returns 1 if the LLM follows the instructions and 0 otherwise.

2.8.6.3. Other Requirements

Additional performance requirements are necessary to ensure the practical applicability of RAG systems in real-world scenarios and to align system behavior with human preferences.

Among these, **Latency** [325] is a critical factor for user experience, especially in interactive applications, such as chatbots and search engines. Latency is defined as the time measurement from the query submission to the completion of the system's response (i.e., including both retrieval and generation phases), whether for a single user query (Single Query Latency) or across an evaluation question answering (QA) dataset. Minimizing latency is essential for maintaining responsiveness and perceived system quality.

2.8.7. The One-Factor-at-a-Time Method

The One-Factor-at-a-Time (OFAT) method [197, p. 4] is a classical experimental design approach used to investigate the effect of individual input factors on a given outcome or response variable. The method consists of selecting a starting point for each factor (baseline configuration set). Then, a single factor is varied systematically over its range while all other factors are held constant. This sequential testing continues until all factors of interest have been individually assessed, allowing the researcher to isolate and observe the marginal effect of each variable independently. Finally, researchers are able to select the optimal combination of factor values based on the obtained results.

While OFAT is straightforward to implement and interpret, it has several limitations. Notably, it fails to account for interactions between independent factors, which may lead to incomplete or misleading conclusions where such interactions are significant. Moreover, OFAT designs are often inefficient, requiring a larger number of experiments to explore the design space compared to factorial or multivariate approaches.

Despite these drawbacks, OFAT remains widely used in practice for (preliminary) exploratory studies or in contexts where resources or experimental control are limited.

We use this method for the evaluation of our retrieval approach and the baselines, as it fits best to our experimental settings and resources, by defining a base configuration and a range of parameter values for each approach (cf. Chapter 13).

2.8.8. Competency Questions

In the context of knowledge representation and semantic technologies, Competency Questions (CQs) are a foundational concept used to define the scope and functional requirements of an ontology or knowledge base [87, 102]. These questions are expressed in natural language and are designed to reflect the types of information that the classification concept and knowledge description approach (e.g., knowledge graphs, ontologies, taxonomies) should be able to represent and retrieve. As such, CQs serve as a practical tool for guiding both the design and evaluation of knowledge representation approaches.

In this work, we utilize CQs to construct a Question Answering (QA) dataset (cf. Section 10.4.3) for evaluating our retrieval approach and the baselines. The questions are formulated based on our question-answering retrieval taxonomy (cf. Section 10.3), allowing us to assess the performance of different retrieval strategies with respect to varying levels of question complexity and different use cases.

3. State of Practice

In this chapter, we present the state-of-practice in Software Engineering (SE) research processes from different user perspectives (e.g., authors and reviewers). This includes approaches for assessing and describing SE research artifacts (cf. Section 3.1) and for persisting and accessing these artifacts (cf. Section 3.2). Finally, we conclude this chapter with a presentation of initiatives that foster the vision of FAIR infrastructures and sustainable research data management (cf. Section 3.3) to emphasize their importance in today's research context.

3.1. Approaches for Assessing and Describing Software Engineering Research

In the following section, we present current practices and approaches for assessing and describing SE research based on its artifacts (i.e., papers and replication artifacts).

First, we present the ACM SIGSOFT Empirical Standards for Software Engineering, which provide checklists of research methods for authors and reviewers (Section 3.1.1). Second, we describe common review quality criteria in the SE research community (Section 3.1.2). Third, we describe two artifact badging systems used in the SE review community for supplementary materials (Section 3.1.3). Fourth, we present de facto classification standards for describing SE research (Section 3.1.4). Finally, we conclude in Section 3.1.5.

3.1.1. ACM SIGSOFT Empirical Standards for Software Engineering

The ACM SIGSOFT Paper and Peer Review Quality Initiative[1], launched in May 2019 and co-chaired by Paul Ralph (Dalhousie University) and Romain Robbes (Free University of Bozen-Bolzano), aims to develop empirical standards [232] for research methods commonly used in SE to improve the quality of papers and establish a more comprehend, reliable, and transparent peer-review process [233]. To achieve this, the initiative develops and maintains structured checklists for authors and reviewers that describe the community's expectations for specific empirical study types (e.g., benchmarking[2], metascience[3], and questionnaire surveys[4]). These checklists label the properties of a publishable paper as *es-*

[1] `https://sigsoft.org/resources/improvingreview.html` [Last accessed on 2025-07-29]

[2] `https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=Benchmarking` [Last accessed on 2025-07-29]

[3] `https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=MetaScience` [Last accessed on 2025-07-29]

[4] `https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=QuestionnaireSurveys` [Last accessed on 2025-07-29]

sential, *desirable*, or *extraordinary*, depending on the methodological context. The standards are designed to function as living documents that are continuously revised to incorporate best practices and to enhance and structure the peer-review processes in SE. To support practical adoption, the initiative also provides an *Empirical Standards Checklist Generator*[5] for authors and reviewers, supporting one-phase and two-phase reviews in conferences and journals. This structured approach aims to address issues of bias and inconsistency in traditional peer-review methods. In June 2021, the International Conference on Evaluation and Assessment in Software Engineering (EASE)[6] became the first conference to officially recommend applying the SIGSOFT Empirical Standards in its peer-review process. A later field experiment at EASE (June 2023[7]) empirically demonstrated that the use of the standards improves inter-reviewer agreement and supports more consistent evaluation. Consequently, these standards represent more than just vague review quality criteria such as "soundness" (i.e., criteria that describe to what extent the paper's contributions address its research questions and are supported by rigorous application of appropriate research methods). We can find similar evidence standards in other research fields as well, e.g., the CONSORT[8] and PRISMA[9] statements in health and medicine, the American Psychological Association's Journal Article Reporting Standards (APA Style JARS)[10], or the WhatWorksClearinghouse[11] in Education.

3.1.2. Review Quality Criteria

In addition, and orthogonal, to the aforementioned ACM SIGSOFT Empirical Standards for Software Engineering, which provide method-specific checklists to guide empirical research reporting and review, the SE research community traditionally relies on more qualitative review criteria in a venue's peer-review process. These criteria are commonly applied across venues to assess the broader scholarly value of a paper submission, often independent of its methodological type. According to Ghezzi [78] (Politecnico di Milano 1863), there should be at least three main qualitative criteria considered for the evaluation of a research paper: *originality*, *significance*, and *rigor*. Additionally, the *presentation* quality, including the *organization* (i.e., structure of the presentation) and *writing style* of a paper should be evaluated as well. However, he also indicates that the assessment while applying these criteria may vary across different reviewers, as these criteria are qualitative, intrinsically subjective, and open to unconscious bias, as also concerned by the ACM SIGSOFT Paper and Peer Review Quality Initiative (cf. Section 3.1.1). In practice, many SE conferences and journals adopt these qualitative criteria in conjunction with quantitative scoring systems for overall recommendations. Typically, reviewers provide a numerical assessment along a predefined scale, ranging, e.g., from "strong accept" to

[5] https://www2.sigsoft.org/EmpiricalStandards/tools/ [Last accessed on 2025-07-29]
[6] https://conf.researchr.org/track/ease-2021/ease-2021-papers#Call-for-papers [Last accessed on 2025-07-29]
[7] https://conf.researchr.org/track/ease-2023/ease-2023-research?#Call-for-papers [Last accessed on 2025-07-29]
[8] https://www.equator-network.org/reporting-guidelines/consort/ [Last accessed on 2025-07-29]
[9] https://www.prisma-statement.org/ [Last accessed on 2025-07-29]
[10] https://apastyle.apa.org/jars[Last accessed on 2025-07-29]
[11] https://ies.ed.gov/ncee/wwc/ [Last accessed on 2025-07-29]

"strong reject", in parallel with textual justifications referencing originality, significance, rigor, and presentation. More recently, with the rise of the Open Science Movement [193], we can observe the adoption of the quality criterion *Verifiability and Transparency* (adhering to the SIGSOFT Open Science Policies[12]) in several high-ranked SE venues (e.g., the ICSE) as well. These developments reflect a shift toward more evidence-based and reproducible research practices, complementing both qualitative judgment and empirical standards in the peer-review process.

3.1.3. Artifact Review and Badging Systems

In 2011, the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) initiated a novel experiment for a major software conference: giving authors the opportunity to submit for evaluation any artifacts[13] that accompany their (accepted) papers. This marked a significant step toward promoting transparency, reproducibility, and reusability in empirical SE. A similar experiment has since run successfully for several more conferences. In the remainder of this section, we present two artifact badging systems applied and commonly used in the SE research review community, particularly in the context of Artifact Evaluation (AE) tracks. These systems provide a standardized framework for recognizing and communicating the quality and availability of research artifacts, thereby contributing to the broader goals of research integrity and Open Science.

ACM Artifact Review and Badging The ACM Artifact Review and Badging is a policy established by the Association for Computing Machinery (ACM)[14] to promote the reproducibility and replicability of research results (cf. Boisvert [32]). This policy (current Version 1.1)[15] outlines a formal review process for *digital artifacts*[16] associated with research papers, such as software systems, (input) datasets, collected raw data, and scripts to run experiments or to analyze results. The review process is designed to assess *Artifacts Available* (i.e., associated artifacts have been made permanently available for retrieval), *Artifacts Evaluated* (functional and reusable), and *Results Validated* (results reproduced and results replicated) to assign standardized badges that reflect the level of artifact quality and openness. These badges serve as visible indicators of a paper's support for reproducible research and help readers and reviewers identify research that adheres to rigorous scientific standards. By promoting the submission and independent assessment of research artifacts, the ACM Artifact Review and Badging Policy aims to improve the integrity of the research ecosystem by ensuring that experimental results can be independently reproduced and verified, thereby promoting transparency and reliability in scientific research and facilitating the reuse of research materials (data and software).

[12] `https://github.com/acmsigsoft/open-science-policies` [Last accessed on 2025-07-29]
[13] `https://esecfse11-aec.cs.brown.edu/` [Last accessed on 2025-07-29]
[14] `https://www.acm.org/` [Last accessed on 2025-07-29]
[15] `https://www.acm.org/publications/policies/artifact-review-and-badging-current` [Last accessed on 2025-07-29]
[16] `https://dl.acm.org/artifacts/dataset` [Last accessed on 2025-07-29]

NISO Recommended Practice on Reproducibility Badging and Definition The National Information Standards Organization (NISO)[17] developed recommended practices titled *Reproducibility Badging and Definitions*[18] (NISO RP-31-2021) that were published in 2021 and are supported by the publisher Institute of Electrical and Electronics Engineers (IEEE). Similar to the ACM Artifact Review and Badging, they aim to support transparency, reproducibility, and trust in scientific research. This framework aims to introduce a set of standardized definitions and badges designed to identify and communicate the reproducibility status of published research across disciplines. It distinguishes between core concepts such as repeatability, replicability, and reproducibility, and provides a controlled vocabulary and visual badges to indicate the extent to which research findings can be independently verified. The recommended practice encourages scholarly publishers, editors, and research communities to adopt consistent terminology and to implement badge-based signaling mechanisms within the publication and review processes. These badges act as visible indicators of the degree of reproducibility support provided by a study, such as the availability of data, code, workflows, or documentation. By fostering a standardized approach to communicating reproducibility, the NISO guideline contributes to improving the integrity, transparency, and accountability of the scientific record. Through the structured application of reproducibility definitions and badges, this initiative aims to reduce ambiguity in reproducibility claims, support informed decision-making by peer reviewers and readers, and ultimately enable greater reusability and independent verification of research outputs. Notably, the International Conference on Software Architecture (ICSA) has adopted the NISO schema within its AE Track[19]. Since 2021, ICSA invites authors of accepted papers to submit associated artifacts for evaluation under criteria inspired by both the ACM Artifact Review and Badging Policy (Version 1.1) and the NISO Recommended Practice. Submitted artifacts of accepted papers may receive badges such as *Open Research Objects* (ORO), *Research Objects Reviewed* (ROR), *Results Reproduced* (ROR-R), and *Results Replicated* (RER).

Comparison The ACM Artifact Review and Badging Policy and the NISO Recommended Practice represent complementary mechanisms to enhance reproducibility in scientific research. While ACM's policy emphasizes hands-on artifact evaluation within computing research, NISO provides a generalizable, cross-disciplinary framework for expressing and signaling reproducibility status. Together, they form an emerging infrastructure that supports transparent, reproducible, and trustworthy scientific artifact publishing.

3.1.4. De Facto Standard Classifications in Software Engineering Research

We can identify two classification schemas as de facto standards for describing SE research w.r.t. its respective *topic* areas. These standards are, in general, used either for tagging a peer-reviewed paper for publication with predefined keywords or as classification schemas in literature reviews to provide an overview of the state-of-the-art by classifying the included papers accordingly.

[17] `http://www.niso.org/` [Last accessed on 2025-07-29]

[18] `https://www.niso.org/publications/rp-31-2021-badging` [Last accessed on 2025-07-29]

[19] `https://icsa-conferences.org/2021/call-for-papers/artifact-evaluation-track/` [Last accessed on 2025-07-29]

ACM Computing Classification System (CCS) The ACM Computing Classification System (CCS)[20] is a poly-hierarchical (i.e., individual topics can belong to multiple parent categories within the classification system) and faceted subject classification schema developed by the Association for Computing Machinery (ACM) to systematically categorize topic areas across the computing discipline. The current version, CCS 2012, organizes subjects within a six-level hierarchical tree, enabling fine-grained classification while also supporting faceted indexing, that is, the ability to assign multiple, orthogonal categories to a single topic, thereby enhancing classification flexibility. This version defines twelve top-level categories, each representing a major subject area within computing. For researchers in SE, the relevant domain is encapsulated under the main subject area *Software and its engineering*. The system supports consistent keyword assignment in scholarly publishing and serves as a basis for thematic organization in literature reviews and bibliometric analyses. The system has gone through several revisions, starting from the first version published in 1964 to the now current version released in 2012. The CCS has been integrated as an ontology into the search capabilities and visual topic displays of the ACM Digital Library[21]. The CCS is comparable to the Mathematics Subject Classification (MSC)[22] in terms of scope, aims, and structure, being used to organize subjects by area.

IEEE The Guide to the Software Engineering Body of Knowledge (SWEBOK Guide) The Guide to the Software Engineering Body of Knowledge (SWEBOK)[23]) defines a comprehensive and structured overview of the foundational knowledge areas in SE. Developed and published by the IEEE Computer Society (IEEE CS), the current version (SWEBOK Guide V4.0, 2024) encompasses 18 knowledge areas, reflecting the evolving scope and complexity of the discipline. SWEBOK V4.0 introduces substantive updates to the body of knowledge by incorporating emerging topics, refining existing content, and removing obsolete subjects. Notably, three new knowledge areas have been added to reflect critical developments in the field: *software architecture*, *software engineering operations*, and *software security*. These additions enhance the guide's relevance to contemporary SE practice and research. As an internationally recognized standard, the SWEBOK Guide serves as a common reference framework, facilitating consistent terminology, shared understanding, and alignment of practices across academia, industry, and certification programs. It supports educational curriculum development, professional training, and competency assessment, and is instrumental in defining the scope and boundaries of the SE discipline.

3.1.5. Conclusion

The presented community-driven or de facto standards and guidelines are valuable tools to ensure collective enhancement of research quality, reproducibility, knowledge organization, and professional scientific practice. While their scopes differ, they share core values of rigor, transparency, openness, standardization, and community trust. Moreover, they implicitly

[20] `https://dl.acm.org/ccs` [Last accessed on 2025-07-29]
[21] `https://dl.acm.org/` [Last accessed on 2025-07-29]
[22] `https://zbmath.org/classification/` [Last accessed on 2025-07-29]
[23] `https://www.computer.org/education/bodies-of-knowledge/software-engineering` [Last accessed on 2025-07-29]

highlight and provide hints for relevant metadata types for realizing (the benefits of) digital transformation in research and, thus, contribute to sustainable RDM. They offer guidance for structuring and documenting research processes and outputs in a consistent and reusable manner. However, these concepts represent only the first initial steps, as they do not cover the main content-related descriptions of research artifacts. In particular, they lack profound descriptive metadata for a holistic structuring approach of SE research artifacts at a statement level. To fully enable automation, discoverability, and reuse, further developments are needed in the direction of semantically rich metadata schemas and domain-specific classifications tailored to the diverse nature of SE research. In this doctoral thesis, we overcome these shortcomings with a novel classification and general data schema, applied in the research field of SWA in Chapter 7.

3.2. Approaches for Persisting and Accessing Software Engineering Research

In the following section, we present established state-of-the-practice approaches and concepts for accessing and persisting SE research based on its artifacts (i.e., papers and replication artifacts). First, we present concepts related to general-purpose web search (Section 3.2.1) and scientific search engines (Section 3.2.2). Second, we provide an overview of version control systems as well as software- and data repositories (Section 3.2.3). Finally, we conclude in Section 3.2.4.

3.2.1. The Evolution of (Web) Search

In the following, we present the evolution of web search, mainly in the context of the most popular search engine in Germany for general purpose, namely the Google Search engine.

The Full-text Era (1993 - 2004) Web 1.0, often referred to as "The Static Web", denotes the initial phase of the World Wide Web[24] (WWW) that was invented by British scientist Tim Berners-Lee in 1989 while working at CERN[25]. This period was defined by static HTML pages, limited interactivity, and a read-only information paradigm. Websites functioned primarily as digital brochures, offering minimal opportunities for user engagement or dynamic content generation. Data transmission was largely unidirectional, from server to client, with early technologies such as the Common Gateway Interface[26] (CGI) and rudimentary JavaScript enabling only basic interactive features. During the 1990s, i.e., in the early days of web search, search engines like Google (as well as other search engines such as AltaVista, Lycos, or Excite) relied mainly on full-text search, which indexed webpages, searched the entire text of web pages, and matched queries using keyword-based algorithms. This enabled web users to search for specific phrases or keywords within the entire content of web pages. This era also marks the beginning of search engine

[24] `https://home.cern/science/computing/birth-web` [Last accessed on 2025-07-29]
[25] `https://home.cern/` [Last accessed on 2025-07-29]
[26] `https://www.rfc-editor.org/rfc/rfc3875` [Last accessed on 2025-07-29]

optimization (SEO) practices, as website owners began to tailor content to improve their visibility in search results. However, by the end of the full-text search era, the limitations of early search algorithms became apparent, particularly in managing the growing volume and complexity of web content, thereby marking the transition to the PageRank Era [212].

The PageRank Era (2004 - 2012) Web 2.0, coined and disseminated by Dale Dougherty[27] and Tim O'Reilly[28], marked a paradigm shift toward a participatory Web, emphasizing user-generated content, social media, and rich, interactive user interfaces. Technological advancements, particularly in Asynchronous JavaScript and XML (AJAX), JavaScript frameworks, and the widespread adoption of mobile computing—were instrumental in enabling these capabilities. Web 2.0 platforms facilitated bidirectional communication and fostered social collaboration, thereby transforming the web from a static information repository into a dynamic, community-driven system. The PageRank Era, spanning from 2004 to 2012, marks a period in the evolution of search engine technology, primarily driven by Google's PageRank algorithm. Developed by Larry Page and Sergey Brin at Stanford University in 1997, PageRank revolutionized the way web pages were ranked in search results by evaluating the number and quality of links to a page [212].

The Knowledge Graph Era (2012 - 2024) Web 3.0, referred to as "The Semantic and Decentralized Web", aims to make the Web more intelligent, context-aware, and machine-readable. It integrates technologies such as semantic web standards[29] (RDF, OWL, SPARQL), AI, NLP, and increasingly, decentralized architectures based on blockchain and peer-to-peer (P2P)[30] protocols. In 2012, Google introduced the Google Knowledge Graph in a blog post[31], representing a major advancement in search technology by enabling the understanding of real-world entities (e.g., people, places, concepts) and their interrelationships. The blog post, "Things, not strings", underscored a transition toward entity-based information retrieval and semantic web standards. By embedding a structured representation of the world into its search infrastructure, Google redefined both the technical design and user experience of general-purpose search.

The LLM and RAG Era (2024 - until today) In May 2024, Google announced major advancements in its search technology through the integration of generative AI, specifically leveraging the custom Gemini model[32] (previously named Bard). This development aims to enhance the efficiency and effectiveness of search processes by providing more comprehensive and contextually relevant results[33].

[27] Dale Dougherty: "Web 2.0 and the Social Context of Learning" in https://projectinfolit.org/docs/smart-talk-interviews/pil-smart-talks-collection.pdf [Last accessed on 2025-07-29]

[28] Tim O'Reilly: "What Is Web 2.0" in https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html [Last accessed on 2025-07-29]

[29] https://www.w3.org/2001/sw/wiki/Main_Page [Last accessed on 2025-07-29]

[30] https://www.rfc-editor.org/rfc/rfc5694 [Last accessed on 2025-07-29]

[31] https://blog.google/products/search/introducing-knowledge-graph-things-not/ [Last accessed on 2025-07-29]

[32] https://ai.google.dev/gemini-api/docs/models [Last accessed on 2025-07-29]

[33] https://blog.google/products/search/generative-ai-google-search-may-2024/ [Last accessed on 2025-07-29]

Conclusion Web search technologies have progressed from simple keyword lookups to context-aware, sophisticated AI-driven systems that "understand" natural language and user intent. These changes reflect broader advancements in computational linguistics, machine learning, and web architecture. As the Web continues to evolve, the future of search lies in achieving a careful balance between AI capabilities, transparency, ethical considerations, and human trust.

3.2.2. Scientific Search Engines

Currently, researchers are required to utilize scientific search engines (i.e., digital libraries and indexers) to obtain and access research papers (documents that are mostly provided as `PDF` files) in the field of SE based on a search query. Therefore, they have to possess information literacy skills "for utilizing the wide range of information tools as well as primary sources in molding information solutions to their problems" [335]. Scientific search engines and databases require structured queries that align with their indexing and retrieval algorithms. Therefore, researchers have to formulate search strings using Boolean logic (AND, OR, NOT, and NEAR), controlled vocabularies, and keyword variation. This scientific task focuses on coordinating search strategies with database functions in order to optimize relevance, accuracy, and hit rates when searching for scientific papers. As there is no dedicated database for the SE research field, researchers can use multidisciplinary search engines or search engines for Computer Science (CS) and the corresponding database search helps them to define a search query for their needs. It is recommended to individualize the search strategy to each scientific search engine.

Commonly used scientific search engines in SE are Google Scholar[34] (since 2004), IEEE Xplore[35] (since 2000), ACM Digital Library[36] (since 1997), Scopus[37] (since 2004), Web of Science[38] (1997 (web), 1960s (predecessor: SCI)), DBLP[39] (Digital Bibliography & Library Project, since 1993), and BASE (Bielefeld Academic Search Engine, since 2004). Thus, most of the current scientific search engines used in SE emerged between the mid-1990s and early 2000s, reflecting the growth of the digital publication era. More recent platforms like Semantic Scholar[40] (by the non-profit Allen Institute for Artificial Intelligence (AI2)[41]; since 2015) and OpenAlex[42] (by the non-profit OurResearch[43]; since 2022) are leveraging AI to enhance search relevance and metadata inference.

[34] `https://scholar.google.de/` [Last accessed on 2025-07-29]
[35] `https://ieeexplore.ieee.org/` [Last accessed on 2025-07-29]
[36] `https://dl.acm.org/` [Last accessed on 2025-07-29]
[37] `http://www.scopus.com/` [Last accessed on 2025-07-29]
[38] `https://www.webofscience.com/wos/` [Last accessed on 2025-07-29]
[39] `https://dblp.org/` [Last accessed on 2025-07-29]
[40] `https://www.semanticscholar.org/` [Last accessed on 2025-07-29]
[41] `https://allenai.org/` [Last accessed on 2025-07-29]
[42] `https://openalex.org/` [Last accessed on 2025-07-29]
The name OpenAlex is inspired by the Library of Alexandria, the ancient institution renowned for its ambition to collect and catalog all human knowledge. The Library of Alexandria is often considered the first major attempt at creating a universal bibliographic catalog. Consequently, the name OpenAlex is a blended word (portmanteau) for "Alex" referring to Library of Alexandria and "Open" referring to open access, transparency, and free availability of scholarly metadata.
[43] `https://ourresearch.org/` [Last accessed on 2025-07-29]

Despite these advantages, Li and Rainer [182] identify severe usability and functionality issues in these engines that hinder their effectiveness for rigorous academic research in the SE research domain. Notably, they document thirteen distinct bugs affecting eight search engines, including problematic behaviors such as improper handling of Boolean operators and hidden search interface elements that obstruct precise query formulation. Beyond bugs, the authors highlight several functional constraints. This includes limited query syntax, inconsistent interfaces, and a lack of programmatic access that restrict researchers' ability to perform complex, automated, and repeatable searches. These limitations pose significant challenges to replicability, completeness, time-effort in SE literature searches.

3.2.3. Version Control Systems and Data Repositories

In current scientific research practice, especially in research fields such as SE where empirical studies often involve artifacts such as software (e.g., source code, experimental scripts) and datasets, the use of dedicated code and data repositories plays a major role in ensuring reproducibility, transparency, and long-term accessibility of research outputs through metadata. The repositories commonly used in this context serve various functions, including version control, persistent identification (e.g., via Digital Object Identifier (DOI)), and compliance with FAIR and Open Science principles.

Code Repositories and Version Control Systems Code repositories and version control systems are essential for managing source code and facilitating collaboration, version tracking, and open access. The most widely adopted platforms include:

- GitHub[44]: A widely used web-based platform for version-controlled code sharing, particularly in academic and open-source communities. GitHub supports collaborative development and integrates with citation tools (e.g., via GitHub-Zenodo integration, see below) to ensure persistent identifiers (DOIs) for software releases.

- GitLab[45]: Similar in functionality to GitHub but often preferred for self-hosted deployments and increased data sovereignty. GitLab is commonly used by academic institutions and research organizations requiring internal access control.

- Mercurial[46]: A free, distributed source control management tool that is implemented in Python and emphasizes performance, scalability, and simplicity. It uses a directed acyclic graph (like Git) to track changes, supporting fast branching, merging, and collaboration.

These platforms offer extensive support for continuous integration, issue tracking, and documentation, which are features increasingly expected in empirical SE research.

[44] `https://github.com` [Last accessed on 2025-07-29]
[45] `https://gitlab.com` [Last accessed on 2025-07-29]
[46] `https://www.mercurial-scm.org/` [Last accessed on 2025-07-29]

Data Repositories Research data repositories (persistent archives) provide structured access to datasets, often with discipline-specific metadata standards and licensing options:

- Zenodo[47] (derived from Zenodotus, the first librarian of the Ancient Library of Alexandria and father of the first recorded use of metadata, a landmark in library history): Developed by CERN and OpenAIRE, Zenodo is a general-purpose repository that supports a wide range of data types, including software. It assigns DOIs to all uploads, enabling citation and compliance with funder mandates and the FAIR principles. All metadata is openly available under a CC0 license. Its integration with GitHub allows automatic archiving of code repositories.
- Figshare[48]: A multidisciplinary repository widely used in academia for sharing datasets, figures, and other supplementary materials. It supports versioning, licensing, and DOI assignment, and is increasingly used in SE research for sharing experimental data.
- Dryad[49]: Primarily used in life sciences and ecology but increasingly relevant across disciplines, Dryad supports peer-reviewed data sharing and is suitable for empirical studies involving human participants or longitudinal software datasets.
- Institutional and domain-specific repositories: Many universities and research consortia provide their own repositories. In the context of SE research, specialized repositories such as Software Heritage[50] focus on long-term archiving and provenance tracking of source code.

Integration and Best Practices In contemporary empirical SE, best practices involve linking code repositories with data repositories using persistent identifiers (e.g., DOIs), providing comprehensive metadata, and adhering to Open Science and licensing standards. Researchers are increasingly expected to accompany publications with openly accessible code and data artifacts, enabling reproducibility studies and secondary analyses.

Conclusion The effective use of code and data repositories is foundational to modern scientific practice in SE. Platforms such as GitHub, Zenodo, and institutional repositories not only support reproducible and transparent research but also fulfill growing requirements from journals, funding agencies, and data stewardship initiatives. The alignment of repository usage with the FAIR principles and Open Science norms is critical for ensuring the long-term value and impact of research artifacts (see Chapter 6).

3.2.4. Conclusion

Comparing the evolution of web search and scientific search reveals two distinct paradigms within the field of information retrieval, each tailored and optimized for different user

[47] `https://zenodo.org` [Last accessed on 2025-07-29]
[48] `https://figshare.com` [Last accessed on 2025-07-29]
[49] `https://datadryad.org` [Last accessed on 2025-07-29]
[50] `https://www.softwareheritage.org` [Last accessed on 2025-07-29]

intents, data characteristics, and retrieval objectives. Web search prioritizes broad accessibility, scalability, and rapid response times to accommodate vast volumes of heterogeneous, often unstructured data. In contrast, scientific search demands high precision, reliability, and domain-specific contextualization, focusing primarily on semi-structured, peer-reviewed literature as well as associated digital artifacts such as software and datasets.

While web search engines have progressively integrated semantic web technologies and advanced artificial intelligence techniques to enhance relevance and user experience, scientific search systems largely continue to depend on traditional keyword-based retrieval methods. This reliance reinforces a predominantly document-centric approach, which may limit more nuanced knowledge discovery and integration across diverse scientific resources. Consequently, there remains a critical need to advance scientific search capabilities by incorporating semantic understanding and AI-driven reasoning tailored to the complexities of scholarly communication that we aim to address in this doctoral thesis by introducing a novel retrieval approach in Chapter 13.

3.3. Vision of FAIR and Sustainable Research Data Management

To promote the vision of FAIR, Open Science (cf. Section 2.1.3), and sustainable RDM, the National Research Data Infrastructure[51] (NFDI) aims to systematically organize, standardize, and make accessible research data across all disciplines in Germany. NFDI currently comprises 26 research-field-specific consortia[52], complemented by one union of consortia, Base4NFDI[53], which is responsible for cross-disciplinary basic services. In addition, NFDI is also expected to participate in the development of international initiatives, such as the European Open Science Cloud[54] (EOSC). Its primary objective is to establish a sustainable and interoperable infrastructure that enables the long-term preservation, accessibility, reuse, and interdisciplinary integration of research data. By fostering collaboration among academic institutions, research organizations, and data centers, the NFDI seeks to improve the quality and efficiency of scientific research through adherence to FAIR principles.

In doing so, the NFDI addresses the growing demand for reliable data management in response to the increasing complexity and volume of research artifacts, including papers, software, and datasets. It facilitates data sovereignty for researchers while promoting Open Science practices and compliance with national and international metadata standards. In conclusion, the NFDI contributes to strengthening the scientific ecosystem (cf. Section 2.1.1) by ensuring that data generated through publicly funded research remain a valuable and accessible resource for future investigations.

To support these objectives, this doctoral thesis contributes to research knowledge description (metadata specification) and knowledge access in SE by tailoring the evaluations to the sub-research field SWA as case example to show the feasibility of our approach.

[51] `https://www.nfdi.de/association/?lang=en` [Last accessed on 2025-07-29]
[52] `https://www.nfdi.de/consortia/?lang=en` [Last accessed on 2025-07-29]
[53] `https://base4nfdi.de/` [Last accessed on 2025-07-29]
[54] `https://open-science-cloud.ec.europa.eu/` [Last accessed on 2025-07-29]

Part II.

Knowledge Description in Software Engineering Research

4. Overview

Part II presents the main contributions of this doctoral thesis, which establish the conceptual foundation for the overall approach KARAGEN (Knowledge Augmented Retrieval And Generation ENgine) described in Part III. Therefore, we address our first research goal, as outlined in Section 1.2. Each of the four contributions in this part addresses a dedicated research question (cf. Section 1.4) along with corresponding validation questions.

First, we introduce methodological foundations and corresponding contributions C1 and C2 that we use throughout this doctoral thesis.

In Chapter 5, we present a novel evaluation approach (C1) for (hierarchical) classification and data schemas based on the GQM paradigm (see Section 2.8.1) to structure the evaluation over three sequential evaluation goals, namely *structure's suitability* (i.e., coverage of classification concepts in the knowledge description approach), *applicability*, and *purpose*.

In Chapter 6, we propose a novel tool-supported documentation process (C2) for the data extraction in literature studies using version control systems and machine-readable formats to foster collaboration, reproducibility, and Open Science principles. With our guideline, we enhance current standard research workflows while extending the established methodological guideline within the Software Engineering (SE) research community (cf. Kitchenham and Charters [147]).

Second, we introduce the first part of the knowledge description module within our KARAGEN approach.

In Chapter 7, we present our knowledge description approach in terms of a novel classification and data schema (C3) to specify research artifacts and knowledge statements reported in scientific papers. Therefore, we rely on a general concept by describing *Findings*, which enhance state-of-the-art, based on their *Validity* and *Evidence*. We instantiate the concept for Software Architecture (SWA) research and classify domain-specific papers in SE according to our schema. In doing so, we reflect on and analyze current state-of-practice in the SWA domain and derive proposals for the community based on our findings.

As the proposed classification concept is instantiated for a domain-specific research field of SE and only reflects a snapshot in time, we introduce a maintenance and general research process (C4) in Chapter 8 to cope with evolving schemas and to track, document, and quantify changes over time. We apply this approach to enhance and refine our classification schema presented in Chapter 7 based on descriptive metadata.

For a research overview of Part II, please refer to Figure 1.3.

5. An Evaluation Approach for Classification and Data Schemas

This chapter builds on the publications "Introducing an Evaluation Method for Taxonomies" [136] published at the ACM Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering (EASE2022) and "Responsible and Sustainable AI: Considering Energy Consumption in Automated Text Classification Evaluation Tasks" [132] published at the 9th IEEE/ACM International Workshop on Green and Sustainable Software (GREENS@ICSE2025). We ask the following research question along with its corresponding validation questions:

RQ I-1: How to evaluate (hierarchical) classification and data schemas in software engineering research, and how to guide researchers through such an evaluation?

VQ I-1.1: How has our proposed evaluation approach been cited according to the citation classification of Kunnath et al. [166], and what is its impact?

VQ I-1.2: How has our evaluation approach been applied in subsequent research?

This chapter introduces a structured evaluation approach comprising three sequential steps (Section 5.3), based on the Goal, Question, Metric (GQM) approach (cf. Section 2.8.1), for assessing hierarchical classification and data schemas. To this end, we define three distinct evaluation goals, align them to different properties and corresponding template questions. These questions guide the investigation and inform the derivation of appropriate metrics. Each goal, based on the evaluation properties and metrics, is further divided orthogonally into two different evaluation strategies to capture different research approaches. This dual-strategy design facilitates operational guidance for the method design with the primary aim of supporting (1) researchers in the design and conduct of their research to evaluate classification and data schemas, and (2) reviewers in the systematic and consistent assessment of such evaluations in Software Engineering (SE). Furthermore, our approach facilitates the assessment of temporal changes in classification and data schemas, including versioning of the associated knowledge representations (cf. Chapter 8).

To demonstrate the feasibility of our proposed approach, we provide a small running example (see the *mug* taxonomy in Section 5.2.1). Additionally, we conduct a meta-analysis based on cited papers (Section 5.4) and inspect the impact of our evaluation approach according to the citation analysis of Kunnath et al. [166].

In this doctoral thesis, we present a hierarchical classification approach to describe and document research artifacts in SE. We apply the proposed evaluation approach (Section 5.3) in the following sections of Part II and Part III to evaluate our contributions and their applications (e.g., knowledge retrieval), as outlined in Section 5.4 by VQ I-1.2.

Consequently, this chapter constitutes the contribution C1.

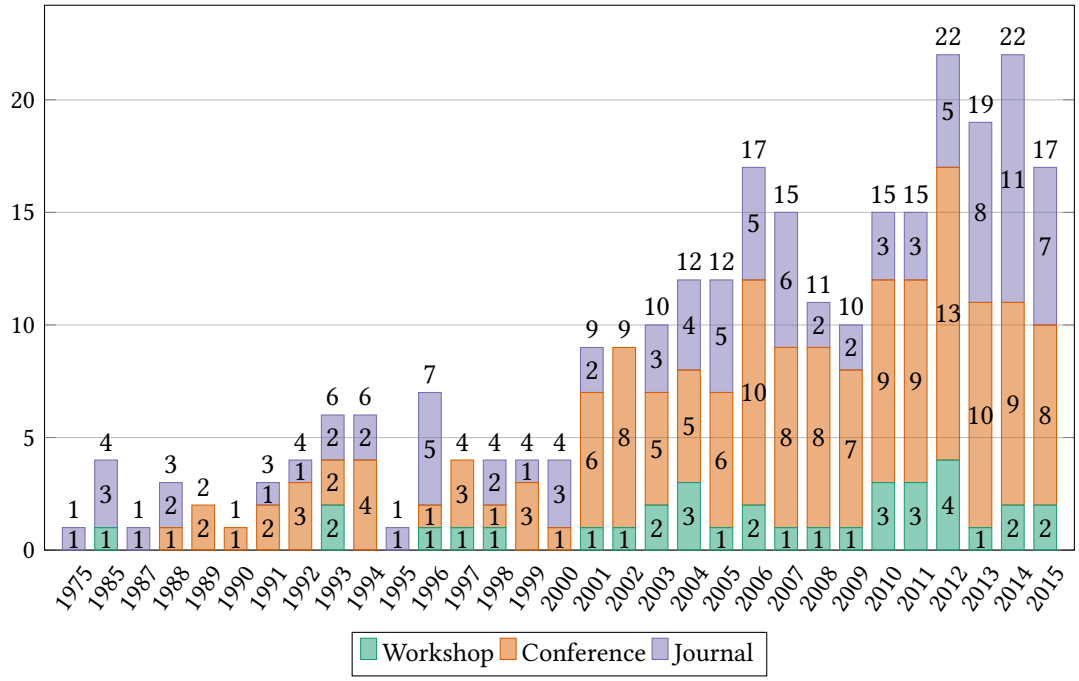

Figure 5.1.: Taxonomies published in software engineering research, distributed by year and publication venue (Figure adapted from Usman et al. [292])

5.1. Background and Problem Statement

A taxonomy[1], also known as a "scheme of classification" according to the Oxford English Dictionary[2], mainly contributes to the organization, collection, and communication of knowledge in science and practice. It can serve to (1) classify objects under study (research objects), (2) provide a common terminology for knowledge sharing, (3) enable a more profound understanding of the interrelationships between the classified objects, (4) identify knowledge gaps and indicate the maturity of a knowledge field, and (5) support decision-making processes [292]. There are various options of classification representations and structures, e.g., a hierarchy (i.e., a hierarchical relationship with inheritance; "is-a" relationship with (or without) mutually exclusive classes), a tree (no inheritance relationship), a paradigm, a facet (multiple perspectives), ring, or network taxonomies like knowledge graphs [292, 231, 24]. In general, based on the most common definitions, they do not always formally specify the properties of and relationships between classes [231]. Taxonomies (or classification schemas) in SE have a practical purpose by categorizing, e.g., processes, approaches, solutions, or meta-research aspects (cf. Section 2.1.1). In our context of addressing research goal (G1), taxonomies serve as classification concepts for metadata of SE research artifacts (papers, data, software), and, thus, offer a valuable approach for representing and structuring research knowledge (see Section 2.5).

As illustrated in Figure 5.1 and supported by Usman et al. [292], there is growing interest in the use and development of taxonomies within SE. However, their study reveals a notable

[1] Taxonomy—gr. *táxis* order, arrangement; *nómos* meaning law or science.
[2] Oxford Dictionary of English, OUP Oxford, 2010. `www.oxforddictionaries.com`

lack of rigorous evaluation practices. Specifically, 45.76% of the reviewed SE literature merely demonstrated the utility of the proposed taxonomy, while 33.58% did not conduct any form of validation.

Evaluating classification schemas remains a complex and inherently challenging task, as indicated in the scientific literature. Despite growing interest, there are no established operational guidelines to support this process. Current approaches and literature surveys (cf. Section 14.1.1) typically enumerate diverse quality attributes, outline generic research workflows, and present research methods in a compendium style. Consequently, they fall short of providing actionable guidance for practical research use. For this, suitable evaluation goals, which investigate quality properties of a classification schema, are crucial. However, these goals and associated properties are often ambiguously defined or inconsistently named in the literature, resulting in a lack of standardized terminology for similar evaluation purposes. Most importantly, these approaches already argue about the relevance of structural properties, utility, and intended purpose, as all of these aspects play a key role in understanding, applying, and comparing classification schemas effectively.

To address these issues and needs, we provide an approach that serves as a practical guide to systematically evaluate classification schemas in SE equally considering the suitability of its structure, its applicability, and its intended purpose in a successive way. Therefore, we followed the GQM paradigm [19] (cf. Section 2.8.1) to construct our approach. We derive our goal-oriented evaluation approach that maps the three mentioned evaluation goals to distinct quality criteria, which are measured by corresponding quantitative or qualitative metrics. Orthogonal and cross-cutting to this alignment, we define evaluation strategies to further classify the underlying research method and method design.

5.2. Preliminaries for the Evaluation Approach

In this section, we introduce a running example (Section 5.2.1) to illustrate the feasibility of our evaluation approach. Further, we clarify the different levels of data and knowledge abstractions regarding the description of digital research artifacts in SE (Section 5.2.2). Finally, we conclude with the different types of classification problems (Section 5.2.3).

5.2.1. Running Example

For our evaluation approach, we consider hierarchical classification and data schemas with categories as nodes (with one root, representing the purpose of the schema) and classes as leaves. In this doctoral thesis, we consider *Digital Research Artifacts* and their abstractions (cf. Figure 5.3) as the primary research object for our evaluations in Part II and Part III.

Our running example to illustrate the feasibility of our evaluation approach represents a hierarchical taxonomy (classification schema) that classifies mugs w.r.t. their size, material, and intended beverage, as depicted in Figure 5.2. This classification schema consists of one root (*Mug*), 3 categories (*Size*, *Material*, *Beverage*), and 8 classes (leaves). The category *Size* is characterized through *Large*, *Normal*, and *Small*. The category *Material* considers *Plastic* and *Ceramic* mugs. The category (hot) *Beverage* comprises *Tea*, *Espresso*, and *Coffee*.

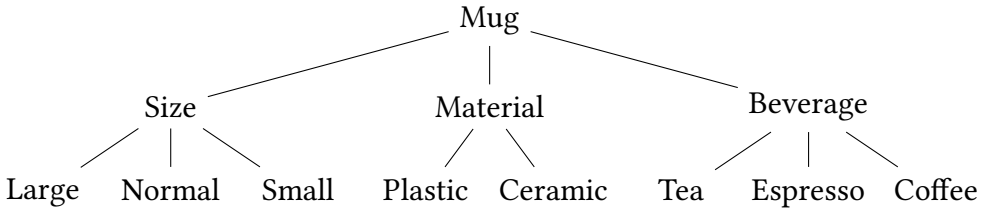

Figure 5.2.: Running example: Classification schema for categorizing different mugs [136]

5.2.2. Levels of Data and Knowledge Abstractions

For understanding the construction and calculation of the metrics in our evaluation approach (see Section 5.3), we have to specify the different levels of data and knowledge abstractions. This structure aligns with the broader and more general concept of the DIKW pyramid introduced in Section 2.4.

In the domain of digital research artifacts and infrastructures, data abstraction is organized into hierarchical layers to facilitate structured knowledge representation, interoperability, and machine-actionability. The abstraction levels M3 to M0 correspond to increasing specificity and decreasing generality in the representation of knowledge and data (in accordance with Figure 5.3 and Section 2.4 / Section 2.5.2.1). Orthogonal to these layers, we derive two dimensions of knowledge description approaches for our concerns: The classification and data schemas (dim1) supporting conceptual approaches, and semantic web technologies, such as KGs (dim2), supporting advanced machine-readable formats. For our evaluations in Part III, we use ORKG (Section 2.6) as (populated) knowledge base and an RDF-based knowledge graph.

The **M3 level** represents the highest abstraction layer, concerned with the meta-meta-model (dim1), which defines the language and formalism used to construct meta-models. It provides the foundational grammar for modeling constructs, such as classes, relationships, and constraints. This is typically instantiated using modeling languages like MOF (Meta Object Facility) or Ecore in the Eclipse Modeling Framework (cf. Section 2.5.2.1). It defines the structure of meta-models themselves, enabling the creation of domain-specific modeling languages. In the context of the ORKG (dim2), the schema at M3 defines the ontological backbone and semantic statements, i.e., the RDF classes, properties, and constraints that govern how knowledge and data are structured (see Section 2.6 and Table 2.3). It ensures semantic consistency and enables reasoning across diverse research fields. In conclusion, the M3 layer ensures that all subsequent modeling layers adhere to a coherent and extensible formalism, enabling semantic interoperability.

The **M2 level** defines the meta-model (dim1), which specifies the structure and semantics of models at the instance level (M1). It includes domain-specific concepts and relationships, serving as a blueprint for creating instance models. Feature diagrams are used to model variability and configuration options of research artifacts. In a broader sense, the meta-model defines the permissible constructs and their interrelationships within a domain (e.g., research objects, research methods, and threats to validity). In turn, ORKG templates (dim2) enhance the ORKG schema by structured forms or patterns that guide users (researchers) in entering research contributions into the ORKG. Thus, they specify which entities and

	Classification and Data Schema	Open Research Knowledge Graph
M3	Meta-Meta-Model Ecore	ORKG Schema (subject, predicate, object)
M2	Meta-Model / Feature Diagram	ORKG templates capturing user-generated ressources
M1	Instance Model: Labeled dataset acc. to M2 schema	Populated RKG
M0	**Digital Research Artifacts** (Papers, Software, Data)	

Figure 5.3.: Different levels of (data) abstraction representing research artifacts with their different knowledge description approaches and corresponding mappings

relations are expected in a given research context and research field. In conclusion, M2 provides the semantic framework for consistent data entry and knowledge-based modeling, enabling the reuse, comparison, and integration of research artifacts and contributions.

The **M1 level** comprises instance models, which are concrete representations of domain-specific data conforming to the meta-model defined at M2 (dim1). This refers to (gold standard) datasets that have been annotated or structured based on the meta-model. For instance, a dataset of scientific papers labeled with knowledge-based metadata such as research objects, research methods, and threats to validity. In ORKG (dim2), this corresponds to the actual knowledge graph populated with research contributions. Each node and edge represents an instance of a concept or relation defined in the template (M2), forming a semantically rich graph of interconnected research knowledge. In conclusion, M1 enables machine-readable and semantically annotated representations of data, facilitating advanced querying, comparison, and data synthesis across research artifacts.

The **M0 level** represents the digital research artifacts derived from research activities (Figure 1.1). They are typically unstructured or semi-structured and require proper modeling to be integrated into higher abstraction layers. In conclusion, M0 artifacts are the basis of scientific knowledge. Their transformation into structured, semantically enriched formats (via M1–M3) is essential for enabling FAIR principles in RDM (Section 2.1.3).

5.2.3. Types of Classification Problems

Digital research artifacts, reflected in the corresponding knowledge description approaches, are complex to describe. For the classification task according to a classification and data schema, we differentiate between three dedicated classification types and different weighted performance scores (cf. Section 5.3.2):

Types of Classification Problems The different types of classification problems determine the complexity of the classification task [158]. We can distinguish between:

- **Binary classification** that refers to the task of assigning each research artifact to one of two mutually exclusive classes based on its metadata. For instance, binary classification is suitable for dichotomous decisions, such as distinguishing between peer-reviewed and non-peer-reviewed artifacts.
- **Multi-class classification** that involves assigning each artifact to exactly one class from a set of three or more mutually exclusive classes. For instance, multi-class classification is suitable for classifying, e.g., the type of research artifact as either a paper, dataset, software, or media (e.g., presentation slides) or other type of (descriptive) metadata such as file format and publication year.
- **Multi-label classification** that allows each artifact to be assigned to multiple classes simultaneously, reflecting the multifaceted nature of research artifacts. For instance, our classification and data schema in Section 7.4 represents such a classification problem.

In **conclusion**, multi-label classification reflects the complex nature of modern digital research artifacts to model them by their descriptive metadata. Thus, it supports richer semantic indexing and enhances the intelligent discoverability of artifacts beyond simple filtering options.

Averaging Performance Scores In evaluation tasks, particularly in classification and multi-label settings or information retrieval tasks, performance scores are often aggregated across classes or instances using macro, micro, and weighted averaging strategies [158]. These approaches provide complementary perspectives on the performance (see performance metrics described in Section 2.8.4), especially when dealing with imbalanced datasets. We can distinguish between:

- The **Macro-Averaged Score** computes the performance metric independently for each class and then averages the results. This approach treats all classes equally, regardless of their frequency in the dataset. This method is particularly informative when assessing performance across all classes, including those that are underrepresented.
- The **Micro-Averaged Score** aggregates the contributions of all classes before computing the metric. It sums the TP, FP, and FN across all classes, thereby reflecting the model's overall performance on individual instances.
- The **Weighted-Averaged Score** computes the metric for each class and then takes a weighted mean, where the weight corresponds to the number of true instances (support) for each class. This method balances the influence of each class according to its prevalence, making it suitable for datasets with significant class imbalance.

In **comparison**, these averaging strategies are crucial for interpreting classification performance in a nuanced manner. The choice among them should be guided by the specific goals of the evaluation and the objective of the real-world application, whether fairness across classes, overall accuracy, or balance between the two is prioritized.

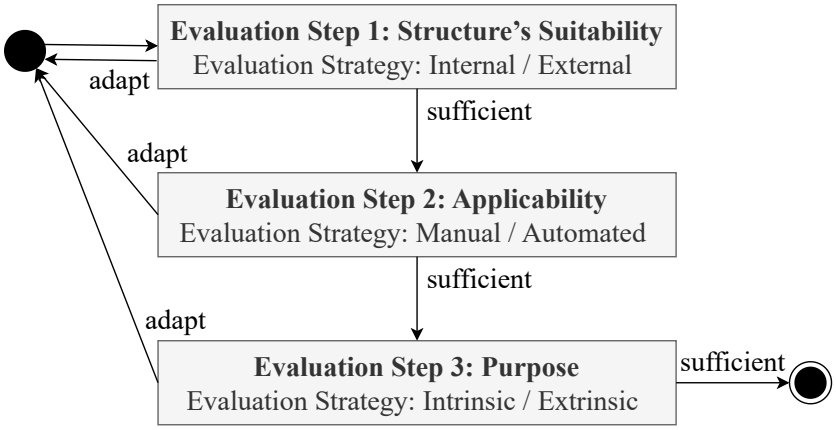

Figure 5.4.: Overview of the process for evaluating classification and data schemas (Figure adapted from Kaplan et al. [136])

Conclusion The choice of a classification type, i.e., binary, multi-class, or multi-label, and suitable weighted performance score depends on the structure and organization of the defined metadata and the intended classification use case. In the context of digital FAIR research infrastructures, such as the ORKG, these classification strategies are instrumental in enabling intelligent knowledge organization and population. As research artifacts require meaningful metadata descriptions to achieve machine-readability, multi-label classification is particularly valuable for capturing the complex semantics of metadata and research artifacts.

5.3. Evaluation Process and Metrics

In this section, we introduce our evaluation approach, while answering our research question *RQ I-1: How to evaluate (hierarchical) classification and data schemas in software engineering research, and how to guide researchers through such an evaluation?*

Scientific literature reveals a wide range of quality criteria for evaluating taxonomies, often described using inconsistent terminology (cf. Section 14.1.1). These criteria have to be systematically considered for an evaluation approach. Accordingly, our approach organizes the criteria across three sequential layers, each supported by orthogonally arranged evaluation strategies, following the GQM paradigm (cf. Section 2.8.1). For each step in the process, we define a specific evaluation *goal*, which is addressed through corresponding *questions* that researchers typically ask in their evaluation. Further, *metrics* are mapped to each quality criterion that aim to answer the dedicated question. Figure 5.4 shows the overall evaluation process.

In **evaluation step 1**, the ***structure's suitability*** of a classification schema is evaluated (cf. Section 5.3.1). Therefore, researchers inspect whether the schema is suitable to classify objects under study (e.g., research papers), by evaluating its generality (level of abstraction), appropriateness (coverage of classification concepts), and orthogonality (of classes). We differ between the internal (author involved in method design) and external (user study)

evaluation strategy. If the structure (regarding the classes as classification concepts) is insufficient, the schema has to be *adapted*, e.g., by defining more sound classes.

In **evaluation step 2**, the ***applicability*** of a classification schema is evaluated (cf. Section 5.3.2). Therefore, researchers inspect whether the schema is usable and yields consistent results when employed by human users (manual evaluation strategy) or machines (automated evaluation strategy). In this way, researchers are able to evaluate reliability, correctness, ease of use, and sustainability (performance and carbon emission trade-off). If the applicability is insufficient (i.e., the schema cannot be applied reliably and correctly enough) the schema has to be improved based on the identified problems and user feedback.

In **evaluation step 3**, the ***purpose*** of a classification schema is evaluated (cf. Section 5.3.3). Therefore, researchers evaluate its relevance, novelty, and significance for the intrinsic evaluation strategy. The latter two quality criteria need at least one preceding schema with the same or a closely related purpose to derive a manual benchmark. If no such preceding schema exists, researchers have to argue why existing classification schemas do not fit. If the *purpose* is insufficient, the schema has to be further extended or an existing schema should be employed instead. For the extrinsic evaluation strategy, researchers evaluate w.r.t. a software application focus, e.g., retrieval system. In this case, the classification and data schema serves as a template to build a knowledge base with population data to assess algorithmic procedures in the AI-based software system.

After any adaptation, the classification and data schema has to be completely re-evaluated, beginning with evaluation step 1 (cf. Figure 5.4). This enables the documentation and tracking of changes, facilitates maintaining an overview of different classification versions and decisions made, and supports the observation of research trends over an extended period (cf. Chapter 8).

5.3.1. Step 1: Evaluating the Structure's Suitability

As a first step, the suitability of the classification's *structure* is evaluated. Since a classification schema should enable the classification of objects under study (e.g., digital research artifacts), it has to exhibit three structural quality criteria: *generality*, *appropriateness*, and *orthogonality*. In this evaluation step, we consider the classes (leaves) of the hierarchical classification schema that refer to terms of the object under study (e.g., scientific paper in level M0) but not the categories (i.e., non-leaf nodes including the root). Within the first evaluation step, we differentiate between two evaluation strategies, namely the internal (Section 5.3.1.1) and external (Section 5.3.1.2) strategy. Researchers may ask the following template questions to operationalize the method design:

> ***Template Questions w.r.t. Evaluation Goal 1 (Structure's Suitability):***
> - To what extent is the «research object» of appropriate <generality / coverage / orthogonality>?
> - How <general> OR <appropriate / appropriately covered> OR <mutually exclusive> are the «classification / descriptive metadata concepts» to describe the <*objects under study*>?

5.3.1.1. Internal Evaluation Strategy

To evaluate and quantify the *generality* and *appropriateness* of a taxonomy, we use the properties for language evaluation as defined by Guizzardi et al. [88] (cf. Section 2.8.2) and adapt the metrics proposed by Ananieva et al. [10]. Although these metrics were originally developed to evaluate a conceptual model w.r.t. tools, we argue that they are also applicable to evaluate the generality and appropriateness of a classification schema. This is because a classification schema, similar to a conceptual model, abstracts a set of objects under study for a specific purpose, and, thus, shares similar structural and functional characteristics that these metrics are designed to evaluate.

Generality (Level of abstraction regarding classification concepts) The metrics *laconicity* and *lucidity* measure the *generality* of a classification schema, i.e., whether it is both general and specific enough to classify certain objects under study.

Definition 5.3.1 (Metrics for Generality)**.** Let C be a taxonomy and $c \in C$ its classes, $\mathcal{R}$ a finite set of objects under study, $R \in \mathcal{R}$ an object under study with relevant terms $r \in R$, and $m_R^C \subseteq C \times R$ a relation between classes $c \in C$ and a relevant term $r \in R$. Then a term $r \in R$ is *laconic* w.r.t. a taxonomy C, iff there is at most one c with $(c, r) \in m_R^C$. The corresponding function $\text{laconic}(C, R, r)$ yields 1 if r is *laconic* and 0 otherwise.

The *laconicity* metric is defined as:

$$\text{laconicity}(C, \mathcal{R}) = \frac{\sum_{R \in \mathcal{R}} \sum_{r \in R} \text{laconic}(C, R, r)}{\sum_{R \in \mathcal{R}} |R|} \qquad \in [0, 1] \tag{5.1}$$

A class $c \in C$ is *lucid*, iff there is at most one $r \in R$ with $(c, r) \in m_R^C$. Conversely, the function $\text{lucid}(C, R, c)$ yields 1 if c is *lucid* and otherwise 0.

The *lucidity* metric is defined as:

$$\text{lucidity}(C, \mathcal{R}) = \frac{\sum_{c \in C} \left(\min_{R \in \mathcal{R}} \text{lucid}(C, R, c) \right)}{|C|} \qquad \in [0, 1] \tag{5.2}$$

Summary. Laconicity determines the fraction of laconic terms among all objects under study, whereas a higher value is better. Low laconicity indicates that a class of a classification schema may be too fine-grained, containing redundant classes that should be merged. Similarly, *lucidity* determines the fraction of lucid classes, ideally approaching one. Low lucidity implies that the classes are too coarse-grained, meaning that there are unspecific classes in the taxonomy that should be split up. In conclusion, a taxonomy demonstrates suitable generality when both laconicity and lucidity values are sufficiently high.

Running Example. To illustrate these metrics, we consider the classification of a *portable cup* using the predefined *mug taxonomy* (Section 5.2.1). One associated term, *suitable for hot drinks*, is not laconic, as it is referenced by multiple classes such as *espresso*, *coffee*, and *tea*. Assuming two other terms are laconic, each uniquely associated with a single class, the resulting **laconicity** is $\frac{2}{3} \approx 0.66$. A low laconicity value suggests that the taxonomy may be overly fine-grained, containing redundant classes that could be merged to improve clarity. Next, we consider the class *plastic* within the mug taxonomy. If this class ambiguously covers distinct materials such as *PET* and *Polystyrene*, it may

not be considered lucid. Assuming all other seven classes are lucid, the resulting **lucidity** is $\frac{7}{8} \approx 0.88$. Low lucidity indicates that certain classes are too coarse-grained and may require subdivision to better reflect the diversity of the objects under study.

Appropriateness (Coverage of classification concepts) Similarly, the metrics *completeness* and *soundness* assess the *appropriateness* of a classification schema, i.e., whether it fully and correctly covers all relevant terms of the objects under study.

Definition 5.3.2 (Metrics for Appropriateness)**.** Under the preconditions of Definition 5.3.1, a term $r \in R$ is *complete*, iff there is at least one $c \in C$ with $(c, r) \in m_R^C$. The corresponding function complete(C, R, r) yields 1 if r is *complete* and 0 otherwise.

The *completeness* metric is defined as:

$$\text{completeness}(C, \mathcal{R}) = \frac{\sum_{R \in \mathcal{R}} \sum_{r \in R} \text{complete}(C, R, r)}{\sum_{R \in \mathcal{R}} |R|} \qquad \in [0, 1] \tag{5.3}$$

Likewise, a class $c \in C$ is *sound*, iff there is at least one $r \in R$ with $(c, r) \in m_R^C$. The function sound(C, R, c) yields 1 if c is *sound* and otherwise 0.

The *soundness* metric is defined as:

$$\text{soundness}(C, \mathcal{R}) = \frac{\sum_{c \in C} \left(\max_{R \in \mathcal{R}} \text{sound}(C, R, c) \right)}{|C|} \qquad \in [0, 1] \tag{5.4}$$

Summary. Completeness denotes the fraction of complete terms over all objects under study. Completeness below the value 1.0 reveals that the schema lacks classes that should be added to cover all relevant terms. By contrast, *soundness* represents the fraction of sound classes in the classification schema. Low soundness indicates that the schema may include unnecessary classes that can be removed, or that the objects under study lack diversity (or lack normative conception). In conclusion, a classification schema is appropriate, if it has both sufficiently high completeness and soundness.

Running Example. To illustrate these metrics, we consider the classification of a *portable cup* using the predefined *mug taxonomy*. One of the associated terms, *closable lid*, is not complete, as it is not represented by any class in the taxonomy. Assuming that the other three terms are complete, each is covered by at least one class, the resulting **completeness** is $\frac{3}{4} = 0.75$. A completeness value below 1.0 indicates that the taxonomy lacks coverage for relevant terms, suggesting that additional classes may be necessary. In contrast, **soundness** measures the proportion of sound classes in the taxonomy. For instance, the class *ceramic* may be considered unsound when evaluating only portable cups, as it does not accurately reflect the characteristics of the object under study (M0 level). If all other seven classes are sound, the resulting soundness is $\frac{7}{8} \approx 0.88$. Low soundness implies that the taxonomy may contain irrelevant or unnecessary classes, or that the diversity of the objects under study is insufficient.

Orthogonality The literature indicates that classes within classification schemas may be either disjoint or overlapping. In this context, Bedford [24] emphasizes that “no two

Table 5.1.: Orthogonality matrix for the mug taxonomy [136]. A value of 1 denotes dependency between classes; a value of 0 denotes independence.

	small	**medium**	**large**	**plastic**	**ceramic**	**espresso**	**coffee**	**tea**	$\sum$
small	-	0	0	0	0	**1**	0	0	**1**
medium	0	-	0	0	0	0	0	0	**0**
large	0	0	-	0	0	0	0	0	**0**
plastic	0	0	0	-	0	0	0	0	**0**
ceramic	0	0	0	0	-	0	0	0	**0**
espresso	**1**	0	0	0	0	-	0	0	**1**
coffee	0	0	0	0	0	0	-	0	**0**
tea	0	**1**	**1**	0	0	0	0	-	**2**
$\sum$	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**4**

categories should overlap or have exactly the same scope and boundaries", thereby advocating for strict orthogonality among classes. To evaluate orthogonality, a self-referencing orthogonality matrix can be employed, where the rows and columns represent the classes of the taxonomy (cf. Ostergaard et al. [211]). Each cell in the matrix is populated with a value of zero if the two classes are independent, or with a value of one if the class in the row implies the class in the column. In some cases, ordinal scales are used to express the degree of dependency between classes. In the literature, orthogonality is also referred to using terms such as synonym uniqueness, mutual exclusiveness, and disjunction.

Running Example. To illustrate the concept of orthogonality in classification schemas, we consider the *mug taxonomy* used to classify various types of portable cups. Orthogonality refers to the degree to which classes are independent and non-overlapping. Table 5.1 presents a simplified orthogonality matrix for the mug taxonomy. In this example, the class *espresso* implies the class *small cup*, and vice versa, indicating a bidirectional dependency. Similarly, the class *tea* implies either a *medium* or *large mug*, whereas the reverse implication does not hold. These relationships may represent either perceived dependencies, based on domain knowledge, or computed functional dependencies derived from the classification of all objects under study [211]. Each cell in the matrix is populated with a binary value: a `0` indicates independence between two classes, while a `1` denotes a dependency, where the presence of one class implies the other. In more advanced settings, ordinal scales may be used to express varying degrees of dependency. In conclusion, fewer dependencies indicate an improved orthogonality for the classification schema and less overlapping leaves (classes).

5.3.1.2. External Evaluation Strategy

For the external evaluation strategy, researchers have to conduct a **user study** based on a survey (questionnaire). This is in extension to the internal evaluation strategy that could invoke bias as the researcher (author) is involved in the mapping process (relevant terms to classification schema).

To assess the granularity of classification concepts and the completeness of a domain-specific classification schema, we propose a survey-based evaluation method involving expert participants. The survey is designed to evaluate each individual class, i.e., each leaf node of the schema, using a **Likert scale** [183] (e.g., a 5-point Likert scale (ordinal scale) with the values *Strongly disagree*, *Disagree*, *Neutral*, *Agree*, and *Strongly agree*), allowing respondents to rate the relevance, clarity, and granularity of each concept. This approach enables the identification of potential gaps, redundancies, or overly fine-grained categories within the schema. Participants are asked to judge whether each class: (i) represents a concept that is relevant to the domain, (ii) is described at an appropriate level of specificity, and (iii) covers the intended semantic scope without overlap or ambiguity.

To systematically quantify the results, researchers can apply the **Content Validity Index (CVI)** [326], a widely used metric in instrument validation. The CVI is calculated at both the item level (I-CVI) and the scale level (S-CVI). I-CVI measures the proportion of experts who rate a given item (class) as relevant (typically using the top two points on a 4-point Likert scale). S-CVI aggregates the I-CVI scores across all items to assess the overall content validity of the schema. A high CVI score indicates strong agreement among experts regarding the relevance and appropriateness of the schema's components. This method supports evidence-based refinement of the classification structure and ensures that the schema aligns with expert knowledge and domain requirements.

5.3.2. Step 2: Evaluating the Applicability

As a second step, the *applicability* of a classification schema is evaluated. Classifications designed for a specific purpose need to be understandable and usable to be applicable [24] by humans and machines. This can be determined and measured by user studies or an automated classification approach (e.g., based on classical or modern AI methods). If user studies (with more than 2 participants) are not feasible, researchers can demonstrate the applicability by themselves using case studies and accept stronger threats to validity regarding the evidence that need to be discussed. Within the second evaluation step, we differentiate between two evaluation strategies, namely the manual (Section 5.3.2.1) and automated (Section 5.3.2.2) strategy, as presented in the following sections. Researchers may ask the following template questions to operationalize the method design:

> ***Template Questions w.r.t. Evaluation Goal 2 (Applicability):***
> - How <applicable / understandable / usable> is the «research object» for <study participants>?
> - How well does an automated classification approach perform on <*datasets*> (in comparison to the baseline/s)?
> - What are suitable configurations regarding performance and energy consumption / carbon emission?

5.3.2.1. Manual Evaluation Strategy

For the evaluation of manual applicability, i.e., when a taxonomy is applied by human participants in a research study, we derive the following quality attributes: *reliability*

(i.e., consistency of results), *correctness* of results (by means of a gold standard), and *ease of use* (as a central usability concept) to cover performance and usability aspects of the classification task.

Reliability To evaluate the reliability of the classification schema, it is essential to demonstrate that different users produce consistent classification results. This is typically done through a user study, in which participants apply the classification to the same set of objects under study. The resulting classifications are then compared between users.

Metrics for Reliability. For this comparison, different metrics can determine the inter-annotator agreement (abbreviated IAA), also known as inter-rater reliability, with at least two participants in a user study. We provide a detailed description of the different IAA metrics, interpretation scales and recommended thresholds in Section 2.8.3

Running Example. In the case of the *mug taxonomy*, individual human participants are assigned the task of classifying each mug from a predefined catalog. Subsequently, the IAA is calculated based on the overall overlap (agreements) of individual classifications for each mug in the catalog.

Correctness Although participants in a user study may produce consistent classification results, these can still deviate from the intended results, especially if category or class definitions are unclear or ambiguous. Therefore, comparing user-generated classifications with a predefined gold standard is essential.

Metrics for Correctness. Metrics such as accuracy, precision, recall, and F_1 score can be used to assess how correctly a classification schema has been applied. Details on the metric definitions are provided in Section 2.8.4. To gain deeper insights, researchers may compute these metrics per class to identify over- or underperformance. For an overall correctness score, we recommend calculating both macro and weighted averages, where the former treats all classes equally, and the latter accounts for class frequency (see Section 5.2.3). This combined view provides a more balanced assessment of the classification's performance.

Running Example. In the case of the *mug taxonomy*, a gold standard can be created by having experts classify a set of mugs from a catalog. This set is then independently classified by multiple users. Based on the comparison between user classifications and the gold standard, metrics such as precision, recall, and accuracy can be computed to assess classification performance.

Ease of Use Besides measuring reliability and correctness, researchers can also assess and evaluate the *ease of use* of a classification schema, indicating whether users can understand and apply the schema easily.

Metrics for Ease of Use. Several methods are available to evaluate ease of use, including questionnaires [100], interviews, and observational techniques such as think-aloud [296].

We recommend applying standardized instruments such as the System Usability Scale [178] and the Computer System Usability Questionnaire (CSUQ) [177]. The SUS consists of a 10-item questionnaire rated on a 5-point Likert scale ranging from "strongly agree" to "strongly disagree." For evaluation purposes, the placeholder <system> in the SUS questionnaire should be replaced with <classification schema> (see Table 5.2). CSUQ,

Table 5.2.: System Usability Scale (SUS) [178]

It.	Intended Question
1	I think that I would like to use this <system> frequently.
2	I found the <system> unnecessarily complex.
3	I thought the <system> was easy to use.
4	I think that I would need the support of a technical person to be able to use this <system>.
5	I found the various functions in this <system> were well integrated.
6	I thought there was too much inconsistency in this <system>.
7	I would imagine that most people would learn to use this <system> very quickly.
8	I found the <system> very cumbersome to use.
9	I felt very confident using the <system>.
10	I needed to learn a lot of things before I could get going with this <system>.

developed by IBM, includes 19 items rated on a 7-point Likert scale to assess user satisfaction with system usability. Both tools are particularly suitable for post-task or post-study evaluations, where users reflect on their experience after completing classification tasks.

Running Example. For our running example (Section 5.2.1), we would simply ask the participants of a user study to complete a usability questionnaire after they conducted the classification task using the classification schema. This post-task evaluation captures users' perceptions of the schema's ease of use.

5.3.2.2. Automated Evaluation Strategy

Besides the assessment of the manual applicability, i.e., applicability by human participants in a user study, the applicability by machines in terms of an automated classification approach, implemented with traditional or modern AI methods in the context of NLP (cf. Chapter 12), can be evaluated. Therefore, research artifacts (M0 level of abstraction) are classified according to a predefined schema (M1 level of abstraction), thereby capturing their descriptive metadata. This constitutes the *Automated Evaluation Strategy* for the second evaluation goal *applicability*. For this, we consider (i) traditional performance metrics, as described in Section 2.8.4 and (ii) the novel combined sustainability metrics [132], as presented in the following.

Performance Quality of an Automated Classifier To evaluate the performance quality of an automated classifier, especially one based on LLMs or other ML techniques, researchers typically rely on a set of well-established metrics. These metrics help assess how accurately and reliably the classifier assigns labels or categories to input data (objects under study).

Metrics for Performance. We provide a detailed overview of performance metrics to assess an automated approach that classifies <objects under study> (e.g., digital research artifacts at the M0 level of abstraction) based on a predefined schema in Section 2.8.4.

Combined Sustainability (Performance / Energy Consumption Trade-off) Besides performance quality, researchers can assess the efficiency. Therefore, we specify our novel sustainability metrics, considering traditional performance metrics (PMs) in combination

with (estimated) carbon emissions derived from the energy consumption (see Section 2.8.5), thereby supporting GreenAI (see Section 2.7). Therefore, researchers have to measure the *total carbon emission (CE)* to evaluate the environmental footprint beside the performance metrics. Best practices and guidelines for the experimental setting are detailed in Section 10.2. Based on carbon emissions, the performance metrics, and the different assets, we derive the following metrics:

Metrics for Combined Sustainability. To relate the carbon emissions to their classification performance (e.g., the macro-averaged F_1 score), Equation 5.5 defines the metric for *relative carbon emission (CE_{rel})*. The metric presents the caused carbon emission per achieved F_1 score, with lower values being better.

$$\mathrm{CE}_{\mathrm{rel}} = \frac{\mathrm{CE}}{f_1} \tag{5.5}$$

Researchers can use a *delta-based version* when comparing different classifiers, as shown in Equation 5.6. To calculate the delta, researchers need to select the worst-performing classifier based on macro-averaged F_1 score as the baseline classifier. Accordingly, $f_{1\mathrm{base}}$ is the worst F_1 score among the classifiers and $\mathrm{CE}_{\mathrm{base}}$ is its corresponding carbon emission. This metric combines the difference in F_1 score with the relation of the baseline's carbon emission compared to the carbon emission of the observed approach. When interpreting the resulting values, higher values are better because the value increases with better F_1 score, better carbon emissions, or both.

$$\Delta_{\mathrm{CE}} = (f_1 - f_{1\mathrm{base}}) \times \frac{\mathrm{CE}_{base}}{\mathrm{CE}} \tag{5.6}$$

In addition to the delta-based version, researchers can use a *normalized version* for assessing different classifiers to simplify the comparison. The *normalized carbon emission ($n(CE)$)* in Equation 5.7 scales the carbon emission between 0 and 1 based on the lowest ($\mathrm{CE}_{\mathrm{lowest}}$) and highest carbon emission ($\mathrm{CE}_{\mathrm{highest}}$) in the evaluation setting. The lowest emitting approach is assigned a 1, the highest emitting approach a 0, and the other approaches are scaled accordingly. The same calculation can be done with $\mathrm{CE}_{\mathrm{rel}}$.

$$\mathrm{n(CE)} = 1 - \frac{\mathrm{CE} - \mathrm{CE}_{\mathrm{lowest}}}{\mathrm{CE}_{\mathrm{highest}} - \mathrm{CE}_{\mathrm{lowest}}} \tag{5.7}$$

In summary, a holistic assessment of automated classifier sustainability has to account especially for performance metrics, energy consumption across different phases (e.g., training and inference), and update cycles (in terms of evolving classification schemas).

5.3.3. Step 3: Evaluating the Purpose

The last step in our approach is concerned with the evaluation of the *purpose*. This includes manual or automated benchmarking to previously existing classification schemas (M2) and their populated variants (M1) of the schema (e.g., in the form of knowledge bases

and datasets). Within the third evaluation step, we differentiate between two evaluation strategies, namely the intrinsic (Section 5.3.3.1) and extrinsic (Section 5.3.3.2) strategy, as presented in the following sections. Researchers may ask the following template questions to operationalize the method design:

> ***Template Questions w.r.t. Evaluation Goal 3 (Purpose):***
> - How <relevant / novel / significant> is the «research object» to former / previous «research objects»?
> - What is the retrieval performance of the <approach> when benchmarked against state-of-the-art approaches on <*datasets*>?
> - What is the answer generation performance of the <*approach*> when benchmarked against state-of-the-art approaches on <*datasets*>?

5.3.3.1. Intrinsic Evaluation Strategy

The semantics of a classification schema and its relationship to previous schemas are evaluated in the intrinsic evaluation strategy (manual benchmark). To this end, the *relevance*, *novelty*, and *significance* are measured for the schema. In this third step, classes (leaves of the schema) and categories (nodes of the schema including the root) of a classification schema have to be considered.

Relevance This evaluation considers whether each individual class and category provides value for the taxonomy's purpose. A taxonomy should only cover classes that are relevant for the objects under study and the taxonomy's purpose, i.e., it should not contain unnecessary or superficial classes. For the mug taxonomy, a category *material density* might not be a valuable addition to help customers of a webshop to distinguish mugs.

Metrics for Relevance. Determining whether a class or category is relevant depends on its specific semantics. Explicitly distinguishing between relevant and irrelevant classes (or categories) enables the calculation of the proportion of relevant elements, which serves as a corresponding metric.

Running Example. In the *mug taxonomy*, classes such as coffee, tea, and espresso may not be relevant for distinguishing mugs, which are typically categorized by functional attributes like suitability for hot or cold beverages. If only 9 out of 12 classes are deemed relevant, the resulting fraction of relevant categories is $9/12 = 0.75$. A lower fraction indicates the presence of irrelevant classes that may require removal to improve the taxonomy's validity.

Novelty While relevance only considers the currently evaluated classification schema, novelty is assessed relative to existing schemas with a similar purpose. It serves as an indicator of the evaluated innovation and adaptation. Innovation reflects the introduction of new classes or categories, whereas adaptation refers to the modification of existing ones to better align with the intended purpose.

Definition 5.3.3 (Metrics for Novelty)**.** Let C be a taxonomy of classes and categories $c \in C$; $\mathcal{T}$ a finite set of previous taxonomies $T \in \mathcal{T}$ with classes and categories $d \in T$;

and $\simeq \subseteq C \times T$ denote that a class/category $c \in C$ is *adapted from* a class/category $d \in T$ (written as $c \simeq d$) whereas $c \neq d$ holds. A class/category $c \in C$ is *new*, if $c \neq d$ and $c \not\simeq d$ for all $d \in T$. The corresponding function $new(C, T, c)$ yields 1 if c is *new* and 0 otherwise.

Then, *innovation* is defined as:

$$innovation(C, \mathcal{T}) = \frac{\sum_{c \in C} \min_{T \in \mathcal{T}} new(C, T, c)}{|C|} \qquad \in [0, 1] \tag{5.8}$$

Similarly, a class/category $c \in C$ is *adapted*, if $c \simeq d$ for any $d \in T$. The corresponding function $adapted(C, T, c)$ yields 1 if c is *adapted* and 0 otherwise.

Then, *adaptation* is defined as:

$$adaptation(C, \mathcal{T}) = \frac{\sum_{c \in C} \max_{T \in \mathcal{T}} adapted(C, T, c)}{|C|} \qquad \in [0, 1] \tag{5.9}$$

Note that $0 \leq innovation(C, \mathcal{T}) + adaptation(C, \mathcal{T}) \leq 1$ holds for arbitrary taxonomies C and finite sets of taxonomies $\mathcal{T}$.

A class or category is considered *adapted* from a previous classification schema if its name, semantics, or position has been modified. It is counted as *new* if no equivalent or adapted version exists in any prior schema. Conversely, it is classified as adapted if at least one previous classification contains a corresponding adapted class or category.

Running Example. For instance, the *mug taxonomy* may be extended by refining the class *plastic* into a category that includes two new classes: *PET* and *PPE*. In this case, the extended classification would yield an *innovation* score of $2/14 \approx 0.14$ and an *adaptation* score of $1/14 \approx 0.07$. The sum of both, *innovation* and *adaptation*, reflects the overall novelty of the evaluated taxonomy. It is important to note that lower innovation and/or adaptation may still be sufficient, depending on the taxonomy's intended purpose.

Significance. A taxonomy is considered more *significant* if it enables a more detailed categorization of the objects under study. To compare taxonomies meaningfully, they must share the same purpose and be applied to a common set of objects. In general, significance is assessed by comparing the number of *equivalence classes* in the evaluated taxonomy with those in previous taxonomies.

Definition 5.3.4 (Metrics for Significance)**.** Let C be a taxonomy of classes and categories $c \in C$; $\mathcal{T}$ a finite set of previous taxonomies $T \in \mathcal{T}$ with classes and categories $t \in T$; $\mathcal{R}$ be a finite non-empty set of objects under study; and $\sim_T \subseteq \mathcal{R} \times \mathcal{R}$ denote an equivalence relation for a taxonomy $T \in \mathcal{T} \cup \{C\}$, whereas $\sim_T$ indicates that a pair of objects is classified identically w.r.t. taxonomy T.

Then, the *classification delta* over $\mathcal{R}$ is defined as:

$$\text{classification delta}(C, \mathcal{T}, \mathcal{R}) = \frac{|\sim_C| - (\max_{T \in \mathcal{T}} |\sim_T|)}{|\mathcal{R}|} \qquad \in [-1, 1] \tag{5.10}$$

The *classification delta* quantifies the normalized difference in the number of equivalence classes between the evaluated taxonomy and the most detailed one. A positive value

indicates that the evaluated taxonomy provides finer distinctions among the objects under study. Conversely, a negative value suggests that a more detailed taxonomy exists and may be preferable. If the delta is zero, the taxonomy may still be sufficiently distinct, but could benefit from incorporating elements of the most significant taxonomy.

Running Example. For instance, the original *mug taxonomy* yields 5 equivalence classes for 5 PET and 5 PPE mugs, treating them all under the general class plastic. In contrast, the extended taxonomy introduces PET and PPE as distinct subclasses, resulting in 10 equivalence classes. This leads to a classification delta of $(10 - 5)/10 = 0.5$. The positive delta indicates that the extended taxonomy enables a more detailed categorization and is therefore considered more significant.

5.3.3.2. Extrinsic Evaluation Strategy

Besides the intrinsic evaluation strategy, we regard the extrinsic strategy to the evaluation goal *purpose* as an antagonist (automated benchmark). The extrinsic evaluation primarily measures the utility of the classification's *purpose* in the context of a downstream application and in real-world scenarios, especially when deploying the classification schema, e.g., in terms of RKGs and AI-based software systems.

In alignment with our **Research Goal 2** (cf. Section 1.2), we focus on the application of the classification schema in QA scenarios, where persisted research knowledge is accessed and utilized to support automated reasoning and information retrieval. To evaluate such systems, specifically RAG systems that incorporate graph-based retrieval mechanisms, we propose a structured evaluation blueprint grounded in the GQM paradigm as well (cf. Chapter 13). This evaluation approach distinguishes between two core evaluation goals: **(1) retrieval quality**, which assesses the system's ability to identify and return relevant and robust knowledge graph segments in response to a query, and **((2) answer generation quality**, which evaluates the accuracy, relevance, and faithfulness of the generated responses based on the retrieved content. Both evaluation goals can be aligned with traditional RAG metrics, LLM-as-a-Judge assessments, or Human-in-the-Loop metrics.

Retrieval quality is particularly sensitive to the structure and variability of the underlying knowledge base (derived from the classification schema). Therefore, we mainly introduce the property of retrieval robustness, which captures the system's ability to maintain consistent performance across different knowledge base variants based on the classification.

Answer generation quality is focused on the key properties: Factual correctness and semantic consistency; answer relevancy, which measures the degree to which the generated response addresses the user's query; instruction following; and answer faithfulness, which assesses whether the response is factually grounded in the retrieved evidence, avoiding hallucinations or unsupported claims.

Central for both evaluation goals is the underlying knowledge base and the **competency questions** which build the bases for a QA gold standard dataset for benchmarking and inherently enable classification schema evaluation. The dataset, in turn, is based on the M2 level of the classification and knowledge base schema. To systematically derive such competency questions, we developed a retrieval **question-answering taxonomy** (see Section 10.3). The taxonomy is based on two orthogonal dimensions: (i) question complexity, which captures the structural and semantic difficulty of retrieving relevant

information (e.g., number of hops, schema reasoning, or entity disambiguation), and (ii) realistic use cases, which reflect practical scenarios in which such systems are deployed based on the metadata concepts in the classification. Each cell in the taxonomy is populated with differently typed questions that serve as competency questions. These questions form the basis for constructing a QA gold standard dataset used in benchmarking. To ensure fairness and avoid bias in evaluation, competency questions are distributed evenly across the taxonomy, enabling a balanced assessment of system capabilities across varying levels of complexity and application contexts. In conclusion, the classification and corresponding populated knowledge bases serve together with the competency questions as evaluation settings. The use case dimension of the taxonomy, in particular, serves as an indicator of the expressiveness of the classification and data schema. As part of the extrinsic evaluation strategy, this evaluation approach supports the validation of the classification's purpose in real-world deployments.

5.3.4. Summary and Limitations

In the previous sections, we presented our evaluation approach. Following, we summarize the answer to our research question RQ I-1 and discuss the limitations of the approach.

We propose a structured evaluation process that follows and adapts the GQM approach (cf. Section 2.8.1) by specifying the structure's suitability, applicability, and purpose of a (hierarchical) classification and data schema as the three main evaluation goals and research objectives in consecutive evaluation steps. Each goal is linked to specific quality attributes (operationalized through template questions) and corresponding metrics, enabling both qualitative and quantitative assessment of the schema quality. Thus, the metrics provide evidence and insights into the quality of a classification schema, indicating the sufficiency of its structure, applicability, and purpose. While it is possible to provide arguments as evidence for some quality attributes, this introduces threats to validity to the evaluation that have to be discussed transparently. Moreover, the evaluation process is designed to be practically applicable and is supported by clearly defined evaluation strategies that are orthogonal to the evaluation goals, properties (questions), and metrics.

Although the current alignment between goals, properties, and metrics supports clarity of the evaluation approach, evolving research practices may introduce new concepts that require integration. Thus, despite its strengths, the approach presents general limitations. These limitations concern the completeness of each element in the evaluation approach, i.e., the specification of the goals, the coverage of quality attributes and metrics as well as the consistency and completeness of evaluation strategies. As such, our evaluation approach should be regularly re-evaluated in line with meta-research principles (see Section 2.1.1).

In conclusion, while the proposed evaluation approach offers a structured and adaptable method for assessing classification schemas, its long-term utility depends on ongoing refinement and validation. Its ability to support self-evaluation and its compatibility with hierarchical schema maintenance make it a promising foundation for future research in schema evaluation and research knowledge representation.

5.4. Application of the Evaluation Approach

To show the feasibility of our approach, we used a running example (cf. Section 5.2.1) for demonstration purpose. Potential impact of our evaluation approach for classification and data schemas can only be assessed over an extended period of time. Nevertheless, we conduct a small citation analysis study to reflect on our evaluation approach in its original publication state in 2022 (cf. Kaplan et al. [136]). Therefore, we analyzed the papers citing our evaluation approach [136] according to the citation classification of Kunnath et al. [166], comprising the classes *Background*, *Compares Contrasts*, *Extension*, *Future*, *Motivation*, and *Uses* (see Table 5.3), addressing our validation question VQ I-1.1. Additionally, we take a closer look at the citation class *Uses*, i.e., analyze papers w.r.t. method design that use our evaluation approach in its original publication state (cf. Kaplan et al. [136]) and its applied extension and usage in this doctoral thesis (cf. Kaplan et al. [137, 131, 132, 134]), addressing our validation question VQ I-1.2.

VQ I-1.1: How has our proposed evaluation approach been cited according to the citation classification of Kunnath et al. [166], and what is its impact?

Since the (original) publication of our evaluation approach in 2022 (cf. Kaplan et al. [136]), it has been cited 22 times (2022: 4 citations, 2023: 7 citations, 2024: 4 citations, and 2025: 7 citations), according to Google Scholar (status: 2025). For the citation analysis, we downloaded the `.bib` file of all results, eliminated duplicates (3 instances) by considering the extended versions, excluded dissertations (3 instances), removed own citations (2 instances), and removed non-English publications (1 instance). As a result, we obtained 13 publications for further analysis (2023: 5 included papers, 2024: 3 included papers, 2025: 5 included papers). We used the documentation process (C2), as described in Chapter 6, by creating a `.taxonomy` file comprising the citation classes according to Kunnath et al. [166] and classified the selected papers. The classification was conducted by two reviewers, who resolved disagreements and ambiguities in the classification tasks. We observed that the classes were not mutually exclusive, resulting in a multi-label classification problem.

The included papers are published across a diverse range of academic publishers: IEEE (2 papers), Springer (2 papers), arXiv (2 papers), and one paper each at Association for Information Systems (AIS), KIT publishing, ACM, Emerald Publishing Limited, Association for Computational Linguistics (ACL), Multidisciplinary Digital Publishing Institute (MDPI), and one unknown publisher. This indicates a great acknowledgment and acceptance of our approach across different research communities beyond SE.

Our classification results show that the most frequent citation class is *Uses* in 7 papers; followed by *Background* and *Compares Contrasts* in 3 papers each; *Motivation* in 2 papers; and *Future* in 1 paper. No paper in the included set was classified under *Extension*, indicating the need for fostering meta-research principles (cf. Section 2.1.1).

Table 5.3 provides an overview of the classification results.

VQ I-1.2: How has our evaluation approach been applied in subsequent research?

To address this validation question, we analyze the cited papers categorized under the citation class *Uses* and demonstrate the application of our proposed evaluation approach within the context of this doctoral thesis.

Table 5.3.: Citation analysis of our evaluation approach [136] based on the citation classification by Kunnath et al. [166]

Citation Class	Class Description	Citation Mapping
Background	Our approach provides background information or is part of the body of literature	[309] , [259] [40]
Compares Contrasts	Citing paper express similarities to or difference from, or disagree (e.g., related work)	[236], [259], [253]
Extension	Citing papers extend the evaluation approach	–
Future	Our approach is a potential avenue for future work	[163]
Motivation	Citing papers are motivated by our evaluation approach	[40], [236]
Uses	Citing papers (partly) apply our evaluation approach	[16], [90], [144], [194], [329], [321], [230]

Based on the results of validation question VQ I-1.1, 7 papers were categorized in the citation class *Uses* (see Table 5.3). Bachoumis et al. [16] (2023) establishes a two-dimensional taxonomy to classify data-driven analytics for reliability in Buildings-to-Grid (B2G) systems. The authors indicate to use our approach by employing all three evaluation steps. Hahner et al. [90] (2023) introduce a taxonomy that categorizes uncertainty in software architecture design specifically in relation to confidentiality. They use all three steps of our evaluation approach. Keim et al. [144] (2023) introduce a fine-grained taxonomy for classifying architectural design decisions, and they evaluate the applicability in terms of reliability. Raja et al. [230] (2024) develop a faceted taxonomy categorizing decision-related knowledge components in bug reports across multiple dimensions relevant for bug prioritization. For the evaluation, they use a combined approach of Ralph [231], [292], and our evaluation approach. Moghaddam et al. [194] (2025) present a hierarchical, faceted taxonomy characterizing abusive language in online communications, derived from an analysis of 18 existing multi-label datasets. The resulting taxonomy, comprising 5 categories and 17 dimensions, is evaluated regarding the structure's suitability and the corresponding properties. Yargan et al. [321] (2025) develop a taxonomy-based evaluation framework to assess existing terminological resources (taxonomies, thesauri, and ontologies) used in biomimetics, while contributing to meta-research and showing that existing approaches lack the necessary consistency, formality, and interoperability. Zhang et al. [329] (2025) introduce a hierarchical, multi-granular taxonomy of factual claims from Social Media, automatically generated using LLMs. In their evaluation, they use the orthogonality and completeness metrics to assess the structure's suitability. In conclusion, we can observe that our approach is used across a broad range of research fields (beyond core SE topics) and with different maturity levels regarding the evaluation steps.

Additionally, we present how we use our (extended) approach for the evaluations of our contributions in Part II (C3, C4) and Part III (C6, C7). Therefore, we mainly regard this doctoral thesis as classified in the citation classes *Extension* and *Uses*. Our contribution C3 constitutes a general classification and data schema for SE research that we instantiate for SWA research by defining SE general categories and domain-specific SWA classes (labels). Therefore, we use evaluation step 1 with the internal strategy and step 3 with the intrinsic strategy. Our contribution C4 constitutes a general research approach and documentation process to maintain evolving classification and data schemas for describing SE research

artifacts based on descriptive metadata. Therefore, we redefine and use all three evaluation steps for this approach. Our contribution C6 constitutes a classification framework. For the evaluation, we use the derived dataset of C3 (data abstraction level M1) and apply the automated evaluation strategy in step 2 by using the performance metrics and the combined sustainability metrics. Our contribution C7 constitutes a retrieval framework. For the evaluation, we use ORKG as populated knowledge graph (data abstraction level M1) based on our data schema in C3 by deriving 4 different template variants (M2) to investigate robustness and the degree of schema-agnostic characteristics of our approach. Further, we use the extrinsic evaluation strategy for the evaluation goal purpose (evaluation step 3) by investigating performance metrics of retrieval and answer generation quality. For the evaluation, we need a QA dataset besides the knowledge bases. We construct this dataset (competency questions) with a balanced distribution across all categories of our question taxonomy to enable the assessment of varying levels of question and retrieval complexity.

5.5. Threats to Validity

In this section, we discuss threats to validity related to our research question (RQ I-1) and validation questions (VQ I-1.1 and VQ I-1.2) w.r.t. the illustrative running example (Section 5.2.1) and the citation analysis (Section 5.4).

External Validity concerns the generalizability of our approach and the results of the citation analysis (application and use of the approach). Our novel proposed approach for evaluating hierarchical classification and data schemas (taxonomies) was originally designed to be generalizable within the context of SE research (i.e., for engineering research). Based on the citation analysis, according to the citation classification by Kunnath et al. [166] presented in Section 5.4, we assume that our evaluation approach is also applicable to other research domains, despite the relatively short time span since its original publication (cf. Kaplan et al. [136]). This assumption is further indicated by the self-validation characteristic of our evaluation approach, which reinforces its potential for broader usage and applicability.

Internal Validity of our evaluation approach is ensured by the granularity of each evaluation step. The selected metrics are adequate and independently focused on either the taxonomy's structural suitability, applicability, or intended purpose. Furthermore, each quality criterion and its corresponding metrics in each evaluation strategy can be independently compiled. The only dependencies arise from the sequential structure of the evaluation process. For instance, a weak assessment of structures' suitability (Step 1) may compromise the validity of the subsequent evaluation of applicability (Step 2).

Construct Validity was established by adopting the GQM approach (cf. Section 2.8.1), which ensures a systematic mapping between evaluation goals, quality properties, and corresponding metrics. The dedicated metrics enable both quantitative and qualitative assessments, thereby supporting robust analysis of evaluation outcomes. The evaluation strategies are orthogonally structured across goals, properties, and metrics, and are

aligned with the characteristics of the method design, ensuring conceptual integrity and methodological consistency.

5.6. Conclusion

This chapter addressed key limitations in the evaluation of taxonomies (classification schemas), namely the lack of operational guidance for researchers and reviewers. In response, we introduced an evaluation approach based on the GQM paradigm (see Section 2.8.1) to structure the process. Accordingly, we defined three consecutive evaluation steps, each aligned with a specific evaluation goal: assessing the *structure's suitability*, *applicability*, and intended *purpose* of a taxonomy. For each step, we define quality criteria (attributes) and corresponding evaluation metrics. Each of these are embedded into a dedicated evaluation strategy to describe the method design. We demonstrated and illustrated the feasibility of our proposed solution using a running example (see Section 5.2.1). Furthermore, we analyzed citations of our approach according to the classification of Kunnath et al. [166] to inspect the evaluation approach's impact and usage (cf. Section 5.4). This analysis provided insights into the adoption and relevance of the approach within the research community. The analysis showed broad coverage according to the citation classes, except for *Extension*. This emphasizes the need to strengthen meta-research practices. In total, 7 papers published in different venues have applied (*Uses*) our evaluation approach to date. This indicates early adoption and potential applicability beyond SE research. In conclusion, our novel proposed and extended evaluation approach in this chapter, based on Kaplan et al. [136, 132, 134], forms the foundation for the evaluations conducted throughout this thesis (see VQ I-1.2), offering a systematic process for assessing classification and data schemas.

6. A Sustainable Documentation Process for Literature Studies

This chapter builds on the publication "A Sustainable Data Extraction Documentation Process for Literature Studies" [135] published at the 3rd International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE@ICSE2026). In the following, we introduce a practical guideline and a generalized, operationalized research documentation process for data extraction during the conduct phase of systematic literature studies. We use this guideline for the documentation of our contributions in Part II. We ask the following research question along with its corresponding validation question:

RQ I-2: How to support open science principles in sustainable literature studies to ensure transparency and replicability in the data extraction phase, especially in a large-scale reviewer (literature review conductors) context, and how to manage such a process?

VQ I-2: How suitable are the selected tools for our proposed solution (RQ I-2) according to the feature analysis set of Marshall et al. [190]?

The principles of the Open Science movement (cf. Section 2.1.3) advocate, among other things, for the open accessibility of scientific research, including results, intermediate steps, and associated artifacts. In Software Engineering (SE) research, however, the availability of essential research artifacts is often insufficient, limiting the maturity and sustainability of scientific research (cf. Santos et al. [252], Konersmann et al. [153], Kazman et al. [143]). Research artifacts, such as papers and replication artifacts, play a crucial role in the sustainable documentation, validation, dissemination, and reuse of research knowledge. Thus, research artifacts as digital outputs of research processes and their corresponding documentation offer substantial value for both researchers and practitioners.

In this chapter, we introduce a practical guideline and documentation process for systematic literature studies that support the data extraction for studies with a categorical objective, i.e., classifying papers according to a predefined or exploratorily developed classification schema derived from metadata specifications. Our approach sustainably (cf. Section 6.2) supports dedicated tasks for the data extraction in a large-scale multi-reviewer context (i.e., involving multiple data extractors participating in the literature study). It aims to complement existing conceptual guidelines (cf. Kitchenham and Charters [147]) by providing instructions for review organizers (leading authors, Section 6.4.1) and data extractors (Section 6.4.2) with dedicated tool support in terms of version control and machine-readable formats (Section 6.3).

In this doctoral thesis, we also use this method for documentation purposes, as employed in Chapter 7 and Chapter 8, and, thus, show the feasibility of our approach, as evidenced

by its acceptance within the respective research community (cf. [155, 137]). Our proposed guideline supports the documentation of the internal and manual evaluation strategy in our evaluation approach (Chapter 5) as well and, thus, enhances the corresponding concept of contribution C1. Furthermore, we conduct a feature analysis (Section 6.5) according to Marshall et al. [190] to assess the suitability of the selected tools incorporated in our guideline. This assessment focuses on the documentation support during data extraction within a collaborative, large-scale reviewer setting.

Consequently, this chapter constitutes the contribution C2.

6.1. Background and Problem Statement

The paradigm of Evidence-based Software Engineering (EBSE) was formally introduced by Kitchenham et al. [150] in 2004 during the 26th International Conference on Software Engineering (ICSE2004[1]). This paradigm is conceptually derived from evidence-based medicine (EBM), which is defined as "*the conscientious, explicit and judicious use of current best evidence in making decisions about the care of individual patients*" [245]. Motivated by the success of EBM in clinical practice, Kitchenham et al. [150] advocated for the adoption of this methodology within the research field of SE. Analogous to EBM, EBSE is focused on collecting and using insights from empirical research, e.g., based on observations and experiments, to determine and evaluate the effectiveness of SE practices, tools, and techniques under specific conditions. Thus, the goal of EBSE is "*to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision-making process regarding the development and maintenance of software*" [150]. Since the introduction of EBSE in 2004, this methodology has gained increasing recognition and adoption within the SE research community. In this context, the most cited methodological guidelines for systematic literature studies in SE are those proposed by Kitchenham and Charters [147] from 2007 that define a research process comprising three phases: planning, conducting, and reporting. Each phase includes a set of defined activities and methodological steps, as listed in Table 6.1.

Systematic literature studies, such as systematic literature reviews (SLRs) and systematic mapping studies, are highly valuable for research and practice, as they provide the best evidence synthesized for dedicated research questions and research objectives. By rigorously collecting, evaluating, analyzing, and synthesizing existing (white and/or gray [74]) literature, they contribute to the development of knowledge bases, the identification of research gaps, and the formulation of future research agendas and roadmaps. However, their conduct is fraught with methodological and practical challenges [252, 39]. One of the most prominent issues is the lack of integrated tool support across the literature study lifecycle fostering increased manual effort and reduced overall efficiency. The data extraction phase is particularly error-prone, as it typically involves manual coding using spreadsheets (such as Microsoft Excel) and the interpretation of study content. This not only introduces inconsistencies but also hampers reproducibility. Moreover, the absence of standardized (data extraction) templates, a reliable knowledge base with supporting information for data extraction, and automation mechanisms further increases the risk of data

[1] `https://dblp.org/db/conf/icse/icse2004.html` [Last accessed on 2025-08-31]

Table 6.1.: Literature review process according to Kitchenham and Charters [147]

Phase 1: Planning the Review	1.1 Identification of the need for a review 1.2 Commissioning a review (optional) 1.3 Specifying research question(s) 1.4 Developing a review protocol 1.5 Evaluating the review protocol
Phase 2: Conducting the Review	2.1 Identification of research 2.2 Study selection 2.3 Study quality assessment **2.4 Data extraction** 2.5 Data synthesis
Phase 3: Reporting the Review	3.1 Specifying dissemination strategy 3.2 Formatting the main report 3.3 Evaluating the report

loss, misclassification, and inconsistent documentation. Collaboration-related challenges also hinder the effectiveness of literature studies. Multi-reviewer studies often suffer from poor coordination, lack of traceability in decision-making, and inconsistent application of data extraction formats. These issues are assembled by lack of management in assigning papers to multiple literature reviewers and tracking the data extraction process, limited support for consensus-building, and lack of version control in existing tools.

With our practical guideline, including version control and tool support, we aim to address these existing challenges and complement existing conceptual guidelines (cf. Kitchenham and Charters [147]). Our approach promotes Open Science principles and supports the development of sustainable literature studies (Section 6.2).

6.2. Sustainable Literature Studies

Santos et al. [252] (2021) aimed at presenting new perspectives on and thoughts about the shift towards more sustainable literature studies. In this context, sustainable literature reviews represent a methodological advancement in how systematic reviews are conducted, maintained, and reused within the SE research community. Traditional literature reviews often suffer from being static, labor-intensive, error-prone, and difficult to update or replicate (cf. Section 6.1). In contrast, sustainable literature reviews aim to address these limitations by promoting practices that ensure long-term usability, transparency, and reproducibility. They address the economic, social, and technical dimensions. The *social dimension* focuses on the human aspects of SLRs, such as communication and collaboration. It ensures stakeholder engagement and open access to results for both academics and practitioners, promoting usability and transparency. The *economic dimension* aims to conserve resources, such as time and human effort, by producing high-quality SLRs. This includes thorough documentation, reliable databases, efficient search tools, and adherence to guidelines to support reproducibility and minimize redundant work. The *technical*

dimension ensures the reliability and usability of tools supporting SLRs. It emphasizes automation and technical support to streamline updates and replications, reducing manual effort. Thus, they define a sustainable literature study as *"the process and a set of actions to make it possible to preserve SLR that endure over time (i.e., longevity) with the least possible time and effort consumed and an effective impact to the industry."* [252]. The core idea is to treat literature reviews not as one-off efforts but as living artifacts that can evolve over time, supporting the aforementioned dimensions. As a consequence, this involves the use of structured (i.e., machine-readable) data formats, version control systems (e.g., Git), and automation tools to support the continuous integration of new evidence. Furthermore, sustainable reviews emphasize openness and collaboration, encouraging researchers to share their protocols, datasets, and analysis scripts in public repositories. By adopting these principles, sustainable literature reviews aim to enhance the scientific rigor, efficiency, and impact of evidence synthesis in empirical research.

With our proposed guideline (cf. Section 6.4) for the data extraction during the conduct phase of literature studies, we particularly focus on supporting these principles in a large-scale multi-reviewer context using machine-readable formats and version control (cf. Section 6.3).

6.3. Tool Support and Machine-readable Formats for Literature Studies

In the following section, we present two tools: the SLR Toolkit (Section 6.3.1) and Version Control Systems (VCSs) (Section 6.3.2). They can support the data extraction phase of literature studies by reducing reviewer errors, to support effectiveness and efficiency, and to contribute to the long-term preservation of extraction results.

6.3.1. The SLR Toolkit

The SLR Toolkit [82] is an open-source, model-driven software framework developed by Sebastian Götz and collaborators to support researchers in conducting SLRs, particularly in the fields of computer science and software engineering. The toolkit aims to streamline and partially automate this complex and often labor-intensive process. It addresses the methodological and organizational challenges of literature studies by offering an integrated environment for managing review workflows, classification schemas, and artifact metadata. The SLR Toolkit supports the following core functionalities:

- Taxonomy Modeling: A key feature of the toolkit allows researchers to design and apply domain-specific (hierarchical) taxonomies. Furthermore, reviewers can assign each paper to one or more taxonomy categories (via checkboxes), with support for automated or semi-automated tagging. This step encourages consistency in coding and facilitates collaborative curation in multi-reviewer scenarios.
- Artifact Management: The toolkit supports the structured storage, annotation, and retrieval of research artifacts, including bibliographic metadata and classification results of reviewers.

- Visual Data Analysis: After classification, the toolkit can generate graphical summaries, such as bar charts, pie charts, and bubble charts. These graphics illustrate distributions across publication years, venues, or user-defined taxonomy nodes.

Technically, the toolkit is primarily implemented in Java, with components in TEX, Swift, HTML, CSS, and Xtend. It leverages technologies such as Xtext for domain-specific language development, JBibTex for bibliographic data handling, and Eclipse BIRT (Business Intelligence and Reporting Tools) for reporting. The build system is Maven-based, and the toolkit is designed for integration with the Eclipse IDE. Distributed under the Eclipse Public License (EPL) 1.0, the SLR Toolkit is available on GitHub[2] and is actively maintained.

The toolkit is especially suitable for our approach, as it enables researchers to define and generate various classification schemas using model-driven engineering techniques, allowing for structured and repeatable categorization of research artifacts. It supports the storage, annotation, and retrieval of bibliographic data and classification results, and facilitates collaborative review processes by enabling independent classification, reviewer assignment, and conflict resolution. Additionally, it includes components for classification and metadata extraction, enhancing scalability and reducing manual effort.

Thus, the SLR Toolkit contributes to the advancement of evidence-based research by enabling reproducible and transparent review processes, supporting the reuse of classification schemas, and bridging methodological rigor with practical usability.

6.3.2. Version Control Systems

VCSs are essential tools in SE practice and modern research. These systems facilitate the systematic management of changes to digital artifacts, including source code, datasets, scripts, and documentation (cf. Section 3.2.3). Therefore, our proposed guideline uses a GitLab instance[3] and its associated features, particularly in the context of literature studies (data extraction phase) and the development of their replication packages.

VCSs are broadly categorized into: (1) Centralized systems (e.g., Subversion[4]), where a single server hosts the repository, and (2) Distributed systems (e.g., Git[5], Mercurial[6]), where each user maintains a full copy of the repository, enhancing resilience and flexibility.

At their core, VCSs provide relevant mechanisms in a research context to:

- Track changes: Every modification to a file is recorded with metadata, including the author, timestamp, and a descriptive message.
- Restore previous versions: Users can revert to earlier states of a file or project.
- Enable parallel development: Multiple contributors can work on different parts of a project simultaneously, with mechanisms to merge changes and resolve conflicts.

[2] `https://github.com/sebastiangoetz/slr-toolkit` (latest release 2024-06-19, version 0.5.0) [Last accessed on 2025-08-24]
[3] `https://docs.gitlab.com/subscriptions/choosing_subscription/` [Last accessed on 2025-08-24]
[4] `https://subversion.apache.org/` [Last accessed on 2025-08-24]
[5] `https://git-scm.com/` [Last accessed on 2025-08-31]
[6] `https://www.mercurial-scm.org/` [Last accessed on 2025-08-31]

- Support branching and tagging: Branches allow for isolated development of features or experiments, while tags mark specific versions, such as releases or publication-ready states.

Git, in particular, has become the de facto standard due to its flexibility, performance, and integration with platforms like GitHub[7] and GitLab[8]. GitHub is a cloud-hosted platform owned by Microsoft[9], primarily offering Software-as-a-Service (SaaS) solutions. It provides a centralized environment for hosting Git repositories, issue tracking, and continuous integration through GitHub Actions. GitLab, in contrast, offers both cloud-hosted and self-managed deployment options. Its open-core model allows organizations to host GitLab on-premises, providing greater control over data, customization, and integration with internal systems. This flexibility makes GitLab particularly attractive for institutions with strict data governance requirements.

In academic research, particularly in empirical SE and reproducibility studies, both platforms serve as repositories for replication packages, datasets, and analysis scripts. However, GitLab's support for self-hosting and group-level templates makes it more suitable for institutional research environments where privacy and reproducibility are paramount. GitHub, due to its widespread adoption and integration with platforms like Zenodo, is often preferred for public dissemination of research artifacts. The ability to assign Digital Object Identifiers (DOIs) to GitHub repositories enhances citation and traceability in scholarly publications.

Both GitHub and GitLab are robust platforms for version control and collaborative development. The choice between them depends on specific project requirements, including deployment preferences, integration needs, and the intended audience for research artifacts. In scientific contexts, GitLab may offer advantages in controlled environments, while GitHub excels in open dissemination and community engagement. Please also refer to Section 3.2.3, where we present state-of-practice in SE research.

For our guideline and proposed research workflow for the data extraction in systematic literature studies, we use GitLab for versioning and tracking (see in the following).

6.4. Our Proposal for a Tool-supported Documentation Process for the Data Extraction

In this section, we introduce our proposal and guideline for a tool-supported literature study, specifically targeting the data extraction in the conduct phase (Table 6.1) for documentation purposes. Our approach is designed to support and manage large-scale review processes, particularly in a context where reviewers are globally distributed. Consequently, we address our research question RQ I-2 by providing structured support for coordination, consistency, transparency, and traceability throughout the data extraction phase. In this way, we enable the tracking of every single intermediate result of the data extraction

[7] `https://github.com/` [Last accessed on 2025-08-31]

[8] `https://docs.gitlab.com/user/` [Last accessed on 2025-08-31]

[9] `https://www.microsoft.com/en-us/thesource-developer/Category/Group/50/github-platform` [Last accessed on 2025-08-31]

according to a predefined classification and data schema. The final results provide input for the next step in the conduct phase, namely the data synthesis. In the context of a double-blind review or to address privacy concerns, all users involved in the literature study need an anonymous Git account with a pseudonym.

In the following, we describe our guideline for a tool-supported literature study to conduct data extraction of included papers from a leading author's perspective (Section 6.4.1) and from a reviewer's (data extractor) perspective (Section 6.4.2).

6.4.1. Our Approach from the Perspective of Leading Authors

The following steps and instructions are described for and from a leading author (researcher) perspective.

Create a Taxonomy File The SLR Toolkit provides the feature to specify a format and data items for the metadata extraction, namely the `.taxonomy` file as an artifact (see Section 6.3.1). The reviewers (literature review conductors) have to follow this specification by selecting and labeling the included papers according to this schema for data extraction.

Create a GitLab Project for the Literature Study The creation of a project[10] within GitLab constitutes the first and foundational step in establishing a collaborative environment. A GitLab project encapsulates, among others, version control, issue tracking, built-in wiki, and access management that we use for our approach. We recommend to create a GitLab organization[11] first to maintain a series of literature studies or to create individual projects for each reviewer (as an alternative for the branches, as described in paragraph 6.4.1). Following, we outline the procedural and conceptual steps for initiating a GitLab project:

To begin, the leading author (with a valid Git account) has to access the GitLab interface and select the option to create a new project. GitLab provides multiple pathways for project creation. In our case, we create a *Blank Project* by initiating an empty repository. During the initialization, researchers define key parameters: *Project Name* (e.g., the name of the literature study) and *Description* (e.g., the research objective of the study); *Namespace*; *Visibility Level* (i.e., Private, Internal, or Public). In addition, one can (and should) select the option to initialize the repository with a `README` file to facilitate immediate interaction. Each GitLab project is backed by a Git repository that serves as the central version-controlled environment for all project assets. GitLab supports granular access control mechanisms that ensure secure and flexible access for both human and machine agents. The GitLab repository should contain the following structure for the literature study:

- A `.bib` file with entries of the papers to investigate that were derived from the study selection and quality assessment process (Phase 2.2 in Table 6.1). Each entry in this file should contain a unique and dedicated BibTeX key identifier with the last name of the first author, the publication year, and optionally a suffix in concatenation according to the format *LastnameYYYYsuffix* (e.g., `Kaplan2025a` and `Kaplan2025b`).

[10] `https://docs.gitlab.com/user/project/` [Last accessed on 2025-08-31]

[11] `https://docs.gitlab.com/user/organization/#create-an-organization` [Last accessed on 2025-08-31]

- A `.taxonomy` file of the SLR Toolkit that facilitates the metadata management and specifies the data extraction format for the reviewers in the literature study.
- A `LICENSE` file under which the repository and data should be published. We recommend to publish the (derived) dataset under the open license `CC BY-SA` (Creative Commons Attribution-ShareAlike[12]).
- A `README` that immediately links to the main page of the respective built-in GitLab Wiki of the repository.

After setting these basic steps, the leading author(s) should create a wiki as knowledge base that we describe in the following paragraph.

Set-up a GiLab Wiki as Knowledge Base and List of Instructions for the Reviewers The built-in GitLab Wiki[13] serves as an information hub where study organizers can provide resources and references to the participants. For instance, we provided information about the whole process including step-by-step manuals via wiki pages. On another page, leading authors should hand out information about the classification schema with a detailed description of each class with examples of the metadata to consider in the data extraction process. Therefore, the wiki serves for the participants (i.e., reviewers in the data extraction phase) as a central place to look for and receive information. For a detailed description of the reviewer process, please refer to our GitLab Wiki[14] of our replication package [155].

Collaboration and Group Integration GitLab projects can be shared within groups to enable team-based work. GitLab supports: *Role-based access*[15] (e.g., Guest, Reporter, Developer, Maintainer), subgroup organization for hierarchical project management, merge request approvals, and issue tracking to enforce structured workflows. The leaders of the literature study have the *Owner role*, reviewers have the *Developer role*.

Repository Branches After preparing the repository with the relevant content, in particular by offering a `.bib` file with all included papers on which data extraction should be conducted, the leading author should create branches[16] for each reviewer. Naming should be clear, we propose the names: `merged` (for final results), `first-reviewer`, `second-reviewer`, ..., `nth-reviewer`, where `n` is the total number of data extractors participating in the literature study. When multiple reviewers work on different entries in the `.bib` file independently and simultaneously, branches prevent (unintentional) conflicts between changes. Thus, each branch acts as an isolated workspace. The default branch (i.e., the main branch) should be and is protected[17] by default to ensure content stability.

[12] `https://creativecommons.org/licenses/by-sa/4.0/deed.en` [Last accessed on 2025-09-26]
[13] `https://docs.gitlab.com/user/project/wiki/` [Last accessed on 2025-08-31]
[14] `https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice/-/wikis/Data-Extraction/Process` [Last accessed on 2025-08-31]
[15] `https://docs.gitlab.com/user/permissions/` [Last accessed on 2025-08-31]
[16] `https://docs.gitlab.com/user/project/repository/branches/#create-a-branch` [Last accessed on 2025-08-31]
[17] `https://docs.gitlab.com/user/project/repository/branches/protected/` [Last accessed on 2025-08-31]

Create Issues / Issue Board and Assign Papers to Reviewers Aside from GitLab serving as a git-server and providing a built-in wiki, it brings additional benefits. Therefore, we use another main feature in our proposed guideline: the extensive issue system[18] with user assignments and labeling. We recommend to create (project) labels, to later categorize issues and track the progress of the data extraction process. Therefore, the leading author should create the following labels:

> *hasFirstReview*, *hasSecondReview*, ..., *hasNthReview*, *merged*, *hasNew*, and *helpNeeded*

For each paper under study, the leading author has to create a separate issue named after the last name of the first author and the year of publication (the BibKey in the corresponding .bib file stored in the repo, e.g., *Kaplan2022*).

Listing 6.1: Example of a `CSV` file for the import in GitLab

```
Title,Description,Issue ID,URL,State,Author,Author Username,Assignee,Assignee
    Username,Confidential,Locked,Due Date,Created At (UTC),Updated At (UTC),Closed
    At (UTC),Milestone,Weight,Labels,Time Estimate,Time Spent

    Review Kaplan2022, Read and classify the paper *Kaplan2022*.
    First Reviewer: Firstname Lastname of first reviewer (@firstReviewer)
    Second Reviewer: Firstname Lastname of second reviewer (@secondReviewer)
    Todos:
    - [ ] if classified by first reviewer, add the label **hasFirstReview**
    - [ ] if classified by second reviewer, add the label **hasSecondReview**
    - [ ] if both reviews are done, discuss your classification results. In case of
        disagreements, please resolve them, commit and push your final results to
        the **merged** branch.
    - [ ] if your results are in the *merched* branch, add the label **merged**

    If you need help, add the **helpNeeded** label to your issue.
    If you have identified a new label relevant to the metadata,
    please add the **hasNew** label to your issue.
    In both cases, we will contact you immediately.

    Review Bucaioni2021, Read and classify the paper *Bucaioni2021*.
    ...
```

There are multiple ways to create issues in GitLab. For instance, one can use `CSV` import by following GitLab's built-in feature that allows bulk issue creation via a properly formatted `CSV` file. The `CSV` import feature in GitLab adheres to a predefined schema, where each row corresponds to a distinct issue and each column represents an attribute of that issue. These (required and optional) attributes comprise, among others, Title, Description, Issue ID, Assignee, Milestone. Thus, they allow for the specification of metadata that governs the visibility, responsibility, scheduling, and categorization of the issue. Listing 6.1

[18] `https://docs.gitlab.com/user/project/issues/` [Last accessed on 2025-08-31]

depicts a template of one entry in the `CSV` file. We recommend following this example format with its content, description and the todo list for the reviewers. To execute the import, the leading author navigates to the Issues section of a GitLab project and selects the `CSV` Import option. Upon uploading the `CSV` file, GitLab parses the content and instantiates the issues within the project repository. Each issue should be (randomly) assigned to a set number of reviewers. With GitLab's issue board[19], one can improve the experience and process for the user and at the same time track the data extraction progress. They offer a visual way to manage and track work and processes within GitLab repositories. Therefore, they provide a structured and visual interface for managing tasks through the organization of issues into columns, referred to as *lists*. Each list typically corresponds to a label, milestone, or assignee, enabling teams (assigned to each paper) to categorize and prioritize work items effectively. We recommend the following lists as different stages in the data extraction: *Open* (all created issues, not yet processed), *helpNeeded*, *hasFirstReview*, *hasSecondReview*, …, *hasNthReview*, *merged*, and *Closed* (with all issues indicating the closing of the data extraction). This system supports agile methodologies by allowing dynamic tracking of issue progress across various stages of the data extraction in the literature study for the leading author(s) and the data extractors.

Documentation of Data Extraction Results For the usage of git, we standardized the commit messages. We provided multiple templates to simplify the usage as well as improve traceability. Such templates for the data extractors are provided in Section 6.4.2.

Support Reviewers In case of any issues that reviewers might face, they can actively request help from the leading authors. This is facilitated with the *helpNeeded* label. Reviewers have to tag the issue (paper) where the problem occurred or create a new issue with this label and assign it to the leading authors. It is now the responsibility of the leading authors to provide appropriate support and guidance to the respective reviewer.

Update `.taxonomy` File if Necessary If a reviewer inspects any missing element / metadata in the taxonomy during the data extraction procedure, they have to inform the leading author by labeling the issue responsible for change with the *hasNew* label. After any update of the `.taxonomy` file, the leading author has to broadcast a message to the reviewers, e.g., via e-mail distribution list or GitLab Broadcast message[20] to pass on all new information.

For the studies in this doctoral thesis that integrate and conduct a literature review in the method design, we use the aforementioned techniques to specify a process for the data extraction phase in a sustainable, documented way.

6.4.2. Our Approach from the Perspective of Data Extractors

The following steps and instructions are described for and from the perspective of data extractors (i.e., reviewers and participants in the literature study).

[19] `https://docs.gitlab.com/user/project/issue_board/` [Last accessed on 2025-08-31]
[20] `https://docs.gitlab.com/administration/broadcast_messages/` [Last accessed on 2025-08-31]

Read Instructions in the Wiki and Set-up the Environment Reviewers should first familiarize themselves with the data extraction process. Therefore, they can inspect the respective GitLab wiki of the literature study, prepared by the leading authors. As a prerequisite, data extractors need to have an anonymous Git account. To start with the data extraction process, reviewers have to download[21], unpack, and run the SLR Toolkit, fulfilling the current system requirements. Further, data extractors have to import the relevant GitLab project repository into the toolkit using Git integration by selecting their dedicated branch (`first-reviewer`, ..., `nth-reviewer`).

Select Review Assignment Each reviewer is (randomly) assigned to specific issues, corresponding to individual and included papers from the study selection (step 2.2 in Table 6.1), by the leading author. To conduct the data extraction process, they have to select an issue from the list, read the corresponding paper, and start the data extraction / labeling process using the SLR Toolkit (see the following description).

Conduct Data Extraction and Labeling For a reviewer, the data extraction begins by selecting an assigned issue from the list in the respective GitLab repository. The corresponding research paper, referenced in the issue, can be accessed in PDF format. Subsequently, the reviewer launches the SLR Toolkit and selects the paper for review. To ensure the latest project data is available in the GitLab repository, the reviewer conducts an update by using the "Team → Pull" function. Within the toolkit, the reviewer navigates to the "BibTeX Entries" tab and enters the BibTeX key (e.g., Mazkatli2020) associated with the assigned issue. Upon locating the entry, the "BibTeX Overview" view is opened for further inspection. Classification is performed using the "Taxonomy" view, where the reviewer selects appropriate leaf nodes to categorize the paper. It is essential to expand non-leaf nodes to access all classification properties / metadata, although only leaf nodes are used for actual classification. The "Questionnaires" view in the toolkit is not required and may be closed. Descriptions of all taxonomy entries are available in the project wiki to support accurate classification. Once the classification is complete, the reviewer saves the results using the "File → Save All" command, which stores the data in the respective `.bib` file.

Version Control and Review Submission Reviewers commit and push their classified labels of a paper and changes using Git version control integrated within the SLR Toolkit to their dedicated branch. Therefore, they publish changes via right click in the n the "Project Explorer" view and select "Team" -> "Commit...". Commit messages follow a standardized format:

```
[IssueRef] [Bibkey] reviewed by [ReviewerName]
```

For instance: `#10 Mazkatli2020 reviewed by A. Kaplan`.

If the changes are updated in GitLab, reviewers have to label the corresponding issue as *hasFirstReview*, *hasSecondReview*, ..., *hasNthReview* according to their respective

[21] `https://github.com/sebastiangoetz/slr-toolkit/releases` [Last accessed on 2025-08-31]

assignment in the issue. Once the reviewers have completed their evaluations, the merging process commences.

Consensus Building and Communication If the reviewers have finished their classification task independently, i.e., an issue is labeled with *hasFirstReview*, *hasSecondReview*, ..., *hasNthReview*, where `N` is the total number of reviewers per issue, they can discuss each classified paper. For the discussion, they can either (1) hold an (online) meeting for synchronous communication, or (2) use the issue's comments for asynchronous communication. To facilitate comparison within the second option, the final classification string can be formatted using a code beautifier (e.g., js-beautify[22]) and shared in the issue comments.

Review Merging for Final Classification Results Upon reaching consensus, the first reviewer (i.e., the leading reviewer of the respective issue) updates the classification via the SLR Toolkit in the *Project Explorer*, commit and push the merged review to the `merged` branch for publishing using the format:

```
[IssueRef] [Bibkey] merged by [ReviewerName]
```

In this format, *[IssueRef]* is the issue Id, e.g., #10, `[Bibkey]` is the key of the entry in the `.bib` file, e.g., Mazkatli2020, and `[ReviewerName]` is the name of the first reviewer (real name or the GitLab account abbreviation). Finally, the first reviewers can mark this issue with the label *merged* to complete their responsible tasks.

After all papers (issues) are classified and resolved by their disagreements, i.e., all labels are `closed`, the data extraction process derives a gold standard dataset and provides input for the data synthesis, the next step in the conduct phase of a literature study. The resulting dataset (export of SLR Toolkit) is a `.bib` file with the entry *classes*, representing the labeling results of the reviewers. Using this structured data file, we developed Visulite (Section 11.4) to provide and generate visualizations of the respective replication package. The tool supports multiple chart types (e.g., bar charts, bubble charts) to enhance the interpretability and representation of the raw data results, and includes a table with filtering capabilities for improved data exploration.

6.5. Feature Analysis and Evaluation

To address aspects of the social, technical, and economic dimensions in sustainable literature reviews, there is a need for the integration of (automated) tool support. Table 6.2 contains a feature analysis according to Marshall et al. [190] to assess and evaluate the suitability of such tools. In this section, we address our validation question VQ I-2.

[22] `https://beautifier.io/` [Last accessed on 2025-08-31]

6.5.1. Feature Analysis Set for Tool Support in Systematic Literature Studies

The feature analysis by Marshall et al. [190] is based on a **set of features** consisting of *Economic* (F1), *Ease of introduction and setup* (F2), *Systematic review (SR) activity support* (F3), and *Process management* (F4). For each feature set, they further derive and decompose into **subfeatures** to enable a granular assessment. Each subfeature is aligned with a specific level of importance: *Mandatory* (M) with multiplier 4, *Highly Desirable* (HD) with multiplier 3, *Desirable* (D) with multiplier 2, or *Nice to have* (N) with multiplier 1. To score the features, a single simple **judgment scale** is defined. A score of 1 is assigned when a feature is fully present or strongly supported. A score of 0.5 indicates partial presence or moderate support, while a score of 0 denotes absence or minimal support. This quantitative assessment facilitates a structured comparison of tool capabilities across the defined feature set. Furthermore, each feature set is aligned with a **weighting score** (see Table 6.2). As previously outlined, a weighted score for each feature is computed by multiplying its raw score by a predefined importance weighting. These weighted scores are then aggregated to derive a percentage score for each feature set, enabling comparative evaluation across tools or configurations.

The percentage score [190] for a feature set is determined as follows:

$$\text{Percentage Score} = \frac{\text{Sum of Weighted Scores}}{\text{Maximum Score}} \times 100 \tag{6.1}$$

The maximum attainable score for a given feature set is defined as the sum of the weighted scores under the assumption that all constituent features are fully present or comprehensively supported. This approach facilitates a normalized assessment of tool performance relative to ideal feature coverage.

To derive an overall performance score for each tool, a weighted average of the normalized scores across all feature sets is computed. Normalization is necessary due to the varying number of subfeatures within each feature set; thus, percentage scores are used to ensure comparability across sets. The weighting schema applied in this calculation is presented in Table 6.2. This schema prioritizes support for systematic SR activities (Feature Set F3) and process management (Feature Set F4), reflecting their critical roles in the context of tool evaluation. Alternative weighting configurations may be adopted depending on the intended emphasis, e.g., to highlight usability aspects as SR tools evolve and mature.

The overall score [190] for a given tool is calculated using the following formula:

$$\text{Overall Score} = \frac{\sum_{i=1}^{n} (w_i TP_i)}{\sum_{i=1}^{n} (w_i)} \tag{6.2}$$

where:

- w_i denotes the weighting assigned to the i^{th} feature set
- TP_i represents the normalized percentage score for the i^{th} feature set
- n is the total number of feature sets considered

Table 6.2.: Feature analysis list for literature study tools [135] acc. to Marshall et al. [190]

Feature Set	ID	Subfeature	Importance	Weighting
F1: Economic	F1-SF01	The tool does not require financial payment to use.	HD (*3)	0.1
	F1-SF02	Maintenance	HD (*3)	
F2: Ease of introduction and setup	F2-SF01	The tool has reasonable system requirements.	M (*4)	0.2
	F2-SF02	Simple installation and setup.	HD (*3)	
	F2-SF03	There is an installation guide.	HD (*3)	
	F2-SF04	There is a tutorial.	HD (*3)	
	F2-SF05	The tool is self-contained.	HD (*3)	
F3: SR activity support	F3-SF06	Data extraction and validation	HD (*3)	0.4
F4: Process management	F4-SF01	Support for multiple users	M (*4)	0.3
	F4-SF02	Document management	M (*4)	
	F4-SF03	Security	D (*2)	
	F4-SF04	Management of roles	HD (*3)	
	F4-SF05	Support for multiple projects	M (*4)	

This approach ensures a balanced and context-sensitive assessment of tool capabilities, accommodating both the structural diversity of feature sets and the strategic priorities of the feature analysis evaluation.

6.5.2. Evaluation Results

We use the aforementioned feature set to assess the selection of our tool support in our proposed guidelines. As this framework for feature analysis of SLR tools focuses on the overall literature study process, we consider F1, F2, and F4 in their entirety and discuss F3 just for the subfeature *Data extraction and validation* (F3-SF06). The following assessment for the requested features is based on the available documentation[23], repository information[24], and the corresponding paper by Götz [82]. For the feature analysis, we assume that leading authors and data extractors use the latest release of the SLR Toolkit and have an account for the used GitLab instance.

VQ I-2: How suitable are the selected tools for our proposed solution (RQ I-2) according to the feature analysis set of Marshall et al. [190]?

To answer this validation question, we discuss each feature set (F1-F4) in the following.

Feature Set 1: Economic This feature set addresses economic factors of the tool related to costs (F1-SF01) and maintenance (F1-SF02).

The SLR Toolkit is publicly available on GitHub under Eclipse Public License 1.0 and seems to be well maintained (latest update 2024-06-19 with 25 releases[25] and 23 contributors in total). A mobile version is available as well. GitLab has three deployment options[26], i.e., GitLab.com (as a cloud-based software-as-a-service (SaaS) platform; hosted by GitLab),

[23] `https://docs.gitlab.com/` [Last accessed on 2015-08-31]
[24] `https://github.com/sebastiangoetz/slr-toolkit` [Last accessed on 2025-08-31]
[25] `https://github.com/sebastiangoetz/slr-toolkit/releases` [Last accessed on 2025-08-31]
[26] `https://about.gitlab.com/pricing/` [Last accessed on 2025-08-31]

Self-Managed, and GitLab Dedicated (single-tenant SaaS for highly regulated enterprises). GitLab.com (proprietary license based on GitLab EE) is free for individuals working on personal and open source contributions, ideally for our guidelines. GitLab CE is fully open source (under MIT License) and maintained publicly. GitLab uses Semantic Versioning[27] for its releases (i.e., major, minor, patch). Its latest release[28] was scheduled by August 21st, 2025 (Version 18.3).

In summary, both tools are free to use and well maintained. Thus, they achieve the highest possible score of 3 points each for feature set F1, resulting in a total of 6 points.

Feature Set 2: Ease of introduction and setup This feature set addresses the usability and initial setup complexity. Each tool should: exhibit reasonable system requirements (F2-SF01), avoiding the need for specialized hardware or software; provide a straightforward installation and configuration process (F2-SF02), supported by an installation documentation (F2-SF03) and/or a tutorial (F2-SF04); function primarily as a self-contained application with minimal dependencies on external technologies (F2-SF05).

The SLR Toolkit exhibits reasonable system requirements (F2-SF01), relying on standard components such as Java Development Kit (JDK) 17 that are commonly used in academic and professional software development environments, and do not impose excessive computational or platform-specific constraints. The installation and setup process (F2-SF02 / SF03) is relatively simple and straightforward (i.e., download, unpack, run `slr-toolkit.exe`). Video tutorials are available, offering a brief overview of the toolkit's functionality and usage. Additionally, there is a paper of the SLR Toolkit published, as aforementioned (F2-SF03 / SF04). The SLR Toolkit is self-contained (F2-SF05). GitLab.com (SaaS) offers a highly accessible and user-friendly environment for collaborative software development. Its system requirements (F2-SF01) are minimal, requiring only a modern web browser or Git client and internet connectivity, which ensures broad compatibility across devices and operating systems. The platform supports immediate use upon account creation, eliminating the need for local installation or configuration, and an installation guide (F2-SF02 / SF03). Consequently, traditional installation guides are relevant only for self-hosted GitLab instances. Instead, GitLab provides extensive documentation and tutorials (F2-SF04), including onboarding materials and interactive learning resources, which facilitate effective user engagement and skill development. Furthermore, GitLab.com is self-contained from the user's perspective. All core functionalities are integrated within the web interface, and no external software dependencies are required (F2-SF05).

In summary, both tools are easy to set up, well documented and technically sound, very user-friendly in terms of documentation and tutorials provided. Thus, they achieve the highest possible score of 16 points for feature set F2, resulting in a total of 32 points.

Feature Set 3: SR activity support This feature set addresses tool support for each activity or step of the three main phases of a systematic literature study (Table 6.1). For our purpose, we focus on the subfeature data extraction and validation (F3-SF06). Therefore, the tool should support the extraction and structured storage of qualitative data through

[27] `https://semver.org/` [Last accessed on 2025-08-31]
[28] `https://about.gitlab.com/releases/` [Last accessed on 2025-08-31]

classification and mapping techniques. Additionally, it must enable the extraction of quantitative data, allowing for the systematic capture of numerical information reported in primary studies. We extend the original definition by incorporating additional subfeatures specifically designed for reviewer settings involving two or more data extractors. This enhancement aims to support sustainable practices in literature studies by improving consistency and collaboration during data extraction. This mainly includes knowledge base and definitions for data to extract, tracking of extraction progress, versioning of intermediate classification results, resolving disagreements in the classification task, and propagation of new information. For the data extraction, we have to regard SLR Toolkit and GitLab in combined use.

The SLR Toolkit provides robust support for both qualitative and quantitative data extraction. It includes taxonomy modeling and classification workflows that allow users to define and apply structured categories to qualitative data extracted from primary studies. The toolkit integrates with JBibTeX and Eclipse BIRT for bibliographic and reporting functionalities, which help in organizing and visualizing quantitative results (bibliographic and quantitative metadata). Thus, it can handle nominal scale well, but quantitative data is rather cumbersome to specify. In comparison to traditional spreadsheets, it supports complex data extraction, especially for hierarchical metadata concerns, without information loss. In addition, typos cannot occur in the classification results due to the preliminary specification of the `.taxonomy` file.

GitLab mainly supports the data extraction through process tracking (issue board) and version control. Classification disagreements can be resolved via issue comments (in asynchronous communication among data extractors).

In summary, both tools combined support Open Science principles for the data extraction phase, while supporting the sustainability dimension for literature studies. The score of the feature is slightly reduced to 0.8, as quantitative metadata is cumbersome to specify and extract via SLR Toolkit and reviewers cannot leave additional comments to the extracted and labeled data, indication improvement for development. In total, feature set F3 obtains 2.4 points.

Feature Set 4: Process management This feature set addresses the management capabilities of a literature review tool. The tool should support collaborative work by allowing multiple users to contribute to a single review (F4-SF01). It must offer secure (F4-SF02) document management for large collections of studies (F4-SF03), enable role-based task assignment (F4-SF04), and support multiple review projects concurrently (F4-SF05). As for the feature set 3, we regard the SLR Toolkit and GitLab (SaaS) in combined use.

The SLR Toolkit is primarily designed for individual use, i.e., does not natively support concurrent multi-user collaboration or user account management. However, collaborative work is possible through external version control systems like Git (F4-SF01). Documents are managed via BibTeX imports and structured metadata. The toolkit supports classification, filtering, and annotation of literature. It enables users to organize and analyze documents systematically within project folders (F4-SF02). There is no built-in authentication, encryption, or access control mechanisms (needed). In a collaborative setting like Git, security aspects are covered (F4-SF03). Role management is not supported (F4-SF04). All

Table 6.3.: Feature set scores and overall scores [135]

	F1 (score out of 6)		F2 (score out of 16)		F3 (score out of 3)		F4 (score out of 17)		Total (score out of 42)		Overall Score
SLR Toolkit	6	100%	16	100%	2.4	80%	17	100%	41.4	98.57%	92.00%
GitLab	6	100%	16	100%							

users have the same level of access and functionality. Differentiated permissions or roles must be handled externally (e.g., via Git) if needed. Multiple literature review projects can be managed by creating separate Eclipse workspaces or project directories. Thus, each project can have its own taxonomy, classification schema, and dataset (F4-SF05).

GitLab supports multiple users with robust collaboration features. Users can be added to groups and projects with defined roles and permissions (F4-SF01 / F4-SF04 / F4-SF05). GitLab repositories can store and version control documents, including markdown files and datasets. Wikis (as knowledge base) and file uploads are supported, making it suitable for managing structured and unstructured documents (F4-SF02). GitLab offers, e.g., audit logs, two-factor authentication, and vulnerability scanning (F4-SF03). GitLab provides detailed role management with predefined roles such as Guest, Reporter, Developer, Maintainer, and Owner. Each role has specific permissions at the project and group levels (F4-SF04). GitLab supports unlimited projects per user or group. Projects are organized hierarchically under namespaces, and users can manage multiple repositories with shared or distinct access controls (F4-SF05).

In summary, both tools combined support process management in all subfeatures, and, thus, achieve the highest possible point in feature set F4.

> ***Summary VQ I-2:***
> *Table 6.2 summarizes the results of the feature analysis. Both tools, either in combined use (in F3 and F4) or considered separately (in F1 and F2), achieved high results (even best scores) in the feature analysis. However, we identified optimization potential and possible development extensions for the SLR Toolkit regarding the data extraction specification and the management and extraction of different types of metadata. Overall, the combined use of the SLR Toolkit and GitLab covers a wide range of necessary features based on the feature analysis of Marshall et al. [190] that are needed to support sustainable literature reviews (see Section 6.2).*

6.6. Conclusion

This chapter addressed key limitations in the current state-of-practice for data extraction of literature studies regarding the integration of sustainability aspects (see Section 6.2) and the adoption of Open Science principles. In response, we introduced a practical guideline for leading authors and data extractors (participants in the literature study), complemented by a tool-based process using version control, issue tracking, and wiki pages. The proposed process supports classification tasks (within a systematic literature study) with more than

two reviewers per data item and extracted metadata in the data extraction form of the objects under study (e.g., research papers). With our novel proposal, we aim to foster reproducibility, efficiency, as well as effectiveness in evidence-based studies that tend to be very time-consuming, labor-intensive, and prone to error. We used and followed this proposal for the studies in this doctoral thesis (Part II). We could show the feasibility of our approach through community acceptance, e.g., within the AE track by receiving an artifact badge for [155]. In addition, the feature analysis according to Marshall et al. [190] confirmed the suitability of our tool-supported approach for data extraction in a multi-reviewer setting. However, in alignment with the vision of transforming infrastructures for scholarly communication, we consider this proposal to be a viable interim solution, particularly w.r.t. the provision and use of simple, easily constructed machine-readable formats.

7. A Classification and Data Schema for Software Engineering Research Artifacts

This chapter builds on the publication "Evaluation Methods and Replicability of Software Architecture Research Objects" [153] published at the 19th IEEE International Conference on Software Architecture (ICSA2022). This publication is summarized in [154] as well. An enhancement of the replication package [155] as a serverless web application, called *Visulite* (Visualization of Literature) [41], is presented in Section 11.4. We ask the following research question along with its corresponding validation questions:

RQ I-3: How to describe and classify research artifacts w.r.t. their findings, validity, and evidence (in software architecture research)?

VQ I-3.1: How general, applicable, and suitable is the data schema for its purpose based on our evaluation approach (see Section 5.3)?

VQ I-3.2: What is the state-of-the-practice in software architecture research w.r.t. our proposed classification and data schema?

VQ I-3-3: What are implications and proposals (recommendations) for the software architecture research community based on our study results?

This chapter mainly presents a novel classification and data schema for categorizing Software Architecture (SWA) research artifacts w.r.t. their specific research objects, with a particular focus on evaluation strategies and the provision of replication packages. The proposed schema is based on our previous work (Section 7.2) and is designed to describe research *Findings* in papers, indicating how they contribute to advance the state-of-the-art, as well as the strength of the *Evidence* supporting their *Validity*. While these classification concepts are rather more general applicable to Software Engineering (SE) research and beyond, we derive a domain-specific instance for SWA research in a pilot study. As a consequence, we contribute to a more comprehensive representation of SWA research knowledge statements and facilitate structured access to this information and data (from research artifacts like papers and replication packages).

To investigate the proposed schema, we conduct an evaluation based on our predefined method (cf. Section 5.3), and, thus, assess its structure, applicability, and the intended purpose (i.e., describing and classifying research artifacts). In particular, we evaluate its feasibility through a systematic literature study of 153 papers to analyze the current state of practice in SWA research (Section 7.7). This analysis includes peer-reviewed full papers of the years 2017–2021 from the flagship conferences in SWA, namely the International

Conference on Software Architecture (ICSA) and the European Conference on Software Architecture (ECSA).

Our main goal is to gain insights into research knowledge representation by systematically analyzing a subfield of SE. By investigating SWA research, we aim to derive specialized insights that may inform the development of a more generalizable approach to structure research knowledge within the broader field of SE (cf. Chapter 8).

Consequently, this chapter constitutes the contribution C3.

7.1. Background and Problem Statement

Researchers present many novel approaches and findings in the research field of SE in papers that are published in conferences, journals, and further venues every year while undergoing a peer-review process (cf. Section 1.1). In particular, empirical research has become an increasingly important part of SE research and practice since (the late) 1990s [285]. This trend is also reflected in SWA research, as Galster and Weyns [72] identified in 2016. In general, empirical research serves as a foundation for ensuring the validity and comparability of novel approaches and findings derived from research results. In this context, the replicability of studies is a critical factor in establishing trustworthiness and contributing to the maturity of a research field.

Given the vast number of research papers published annually (see the description of problem statement 2 in Section 1.1), it is essential to understand how research is evaluated to assess the validity of findings and compare related approaches and their results. A structured and comprehensive representation of research knowledge can facilitate this understanding. Moreover, insights gained from a systematic, top-down or bottom-up analyses of research knowledge descriptions can start discussions on the standards that a research community aims to establish (cf. core concepts of meta-research in Section 2.1.1). However, no such overview or knowledge description approach (beyond the categorization of research and evaluation methods) currently exists for SWA research.

To address this gap, we propose a classification and data schema for SE research that we instantiate for SWA research, building upon our previous work (Section 7.2). The schema is initially defined in a top-down manner, with concrete instances specified accordingly, and extended in a bottom-up fashion as needed during the data extraction phase of our systematic literature study. The results and insights gained from this study serve as the foundation for the conceptual approach presented in the following Chapter 8, contributing to a maintainable approach for research knowledge representation in SE.

7.2. Previous Work

The publications “A Classification for Managing Software Engineering Knowledge” [139] published at the 25th International Conference on Evaluation and Assessment in Software Engineering (EASE2021) and “Towards Managing and Organizing Research Activities” [138] published at the 8th ACM Celebration of Women in Computing (ACM womENcourage 2021) form the basis for this section and research question RQ I-3.

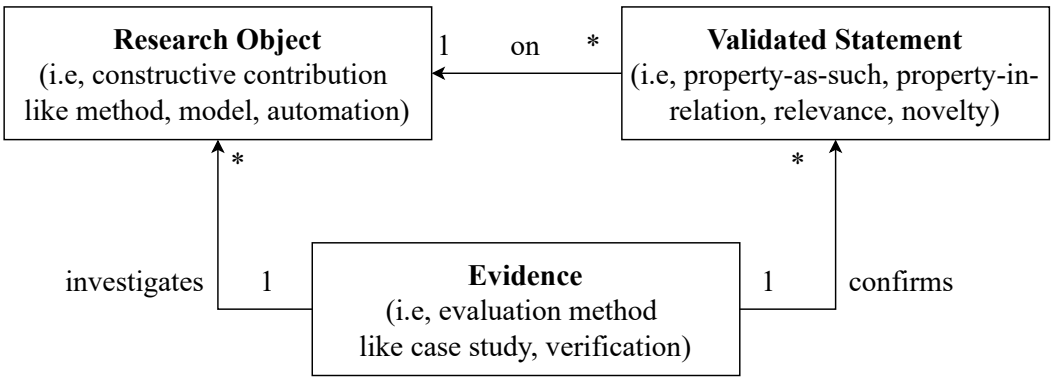

Figure 7.1.: Structured knowledge representation of the main content of a scientific (software engineering) paper, categorized by *Research Object*, *Type of Statement*, and *Evidence* according to Kaplan et al. [139]

Unlike traditional paper-wise classification in SE, our approach adopts a modular concept to describe research knowledge of papers according to their individual statements, highlighting their validity and evidence. In particular, each paper is associated with at least one *Research Object* (i.e., the object under study) and a set of *Validated Statements* about the *Research Object* (properties of the research object), each substantiated by empirical or non-empirical *Evidence*. Additionally, we outlined the extension of this approach in another publication [138]. For this, each combination of the aforementioned descriptive metadata classes should be assigned to corresponding templates for research questions (that operationalize the method design), instructions for the method design (e.g., choosing appropriate metrics according to statement and underlying evidence), and guidelines for reporting validity threats, which need to be discussed according to the research object, the kind of statement, and the underlying evidence. According to our work [139], a *Validated Statement* is defined as a fact, a hypothesis, or conclusions derived from evaluation results related to a given *Research Object* and *Evidence*. These statements can describe (i) an intrinsic property of the *Research Object*, (ii) a relationship between the *Research Object* and its context (e.g., the technical environment or development processes), or (iii) the relevance and novelty (originality) of the *Research Object*.

This statement-based classification paradigm allows for several key relationships between specified research artifacts and their knowledge statements [139]:

- A paper can contain multiple *Validated Statements* (about *Research Objects*).
- Each statement is supported by different types and varying strengths of *Validity* and *Evidence*.
- Follow-up papers may contribute additional validity and evidence to existing statements from prior research, either by providing new (empirical) data, through verification, or by strengthening arguments.
- Papers may introduce statements that refine those made in earlier studies.
- Papers may also present statements that contradict prior findings, provided they are supported by appropriate evidence.

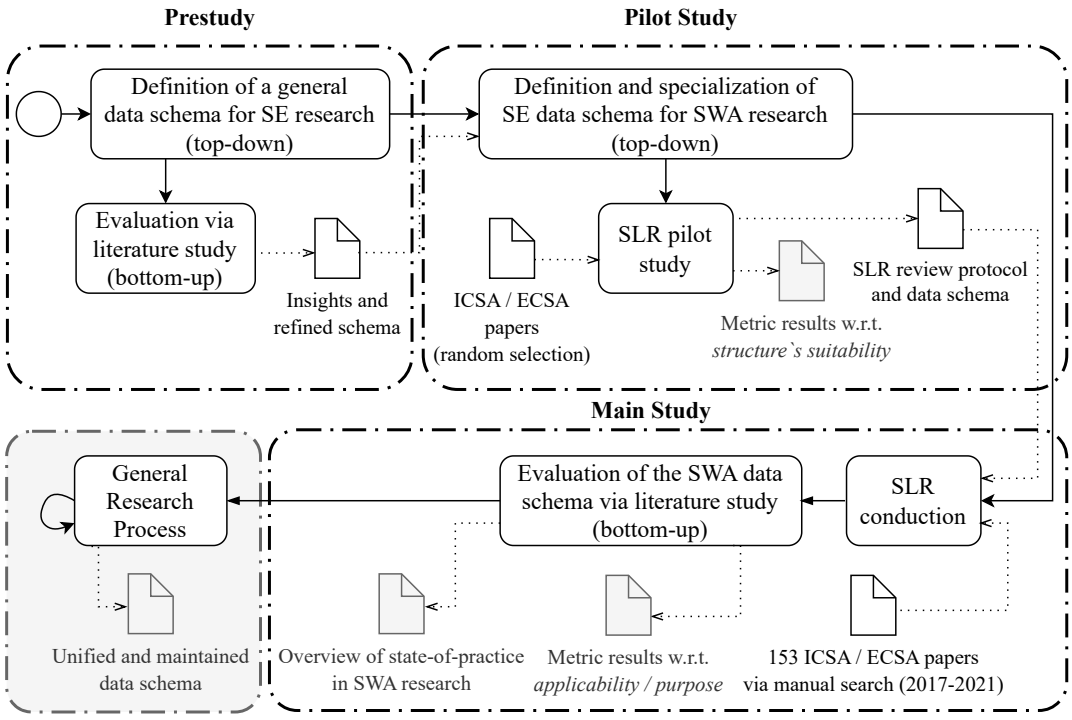

Figure 7.2.: Research approach for deriving and evaluating the classification and data schema for software architecture research

Figure 7.1 depicts the aforementioned classification concepts and their relationships. This classification and data schema was evaluated via a literature review and in a small user study [138]. Based on these insights, we refined and improved the schema.

7.3. Research Approach

Our research approach consists of four phases: (1) a prestudy (cf. previous work as presented in Section 7.2), (2) a pilot study (to derive the research approach for the main study), (3) a main study (cf. Section 7.2), and (4) a unified approach and general research process to cope with knowledge description approaches and data schemas and their changes over time (cf. Chapter 8). Figure 7.2 depicts these phases with highlighted artifacts, denoting the research results in this chapter. Our previous work [139] provided insights and the basic construction for the classification and data schema (for SWA research). Building on these insights, we refine the schema to capture the specific characteristics necessary for systematically describing research findings, their validity, and evidence in this research domain. In the pilot study, the resulting schema is then used to:

- Evaluate its structure, applicability, and purpose using the evaluation approach outlined in Section 5.3. This evaluation is conducted on a selected set of papers from ICSA and European Conference on Software Architecture (ECSA).

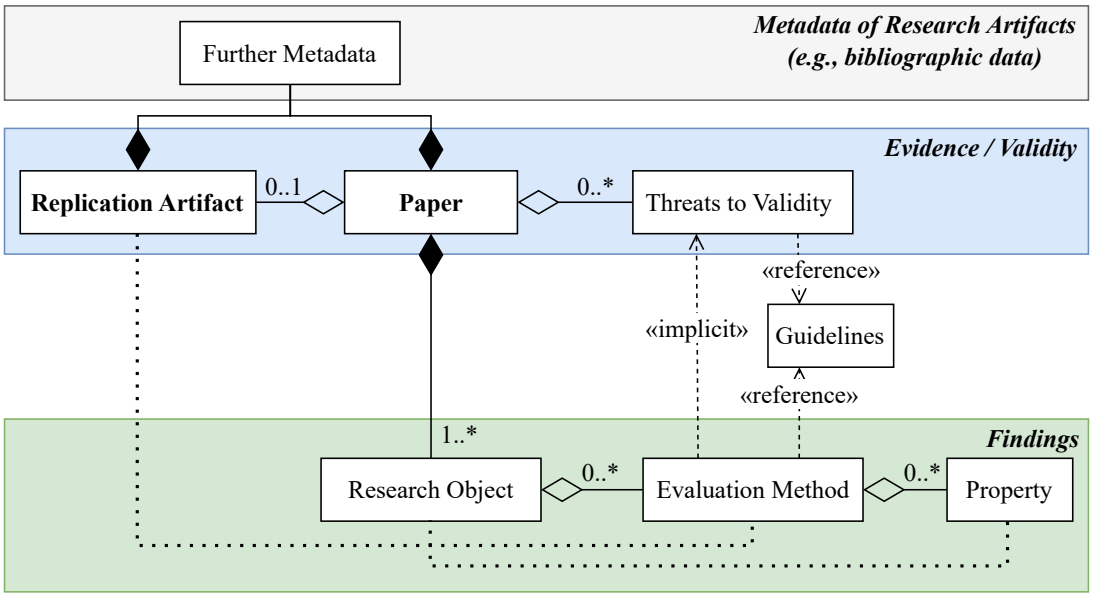

Figure 7.3.: Schema of the data extraction for describing software engineering research papers and replication artifacts. Dashed arrows indicate inferred relations. (Figure extended from [153])

- Develop a systematic literature review (SLR) protocol to guide the main study and evaluate the applicability (cf. evaluation step 2 in Section 5.3).

The main study involves a large-scale evaluation of the finalized SWA research schema through an extensive literature study. In total, 153 papers published at ICSA and ECSA are classified according to the schema by at least two independent reviewers. After this review phase, the reviewers discuss any disagreements to derive and establish a gold standard dataset that serves as the prerequisite for the knowledge base and as the central item for the evaluations in Chapter 12 and Chapter 13. Based on the evaluation step 2 of our method in Section 5.3, we assess the (manual) applicability of the schema. Additionally, we provide an overview of the state-of-practice in SWA research, informed by the classification results. The schema of the main study serves as input for the unified approach (cf. Chapter 8), enabling its maintenance and evolution for future developments.

7.4. A Classification and Data Schema for Software Architecture Research

In this section, we present our approach for describing knowledge statements in SE research artifacts using descriptive metadata. We instantiate this approach for SWA research by answering our research question (RQ I-2): *How to describe and classify research artifacts w.r.t. their findings, validity, and evidence (in software architecture research)?*

Building on our previous work (cf. Section 7.2), we regard a publication on a modular basis, treating them as coherent and self-contained research units to describe knowledge statements based on their findings. Accordingly, the focus of our approach is the *Research Object* (i.e., the object under investigation). Each paper must have at least one Research Object. Within the domain of SWA research, we identify a total of *13* distinct Research Objects (cf. Figure 7.4) in the pilot study.

A Research Object can be evaluated w.r.t. a *Property*, which refers to a specific characteristic of the Research Object under investigation. A Property is currently assigned to three major high-level categories: *Quality in Use* (comprising 5 classes), *Product Quality* (comprising 8 classes), and *Quality of Analytical Method* (comprising 9 classes). These categories can be further subdivided into concrete classes, such as *Effectiveness* and *Efficiency* for *Quality in Use*. Each Property is assessed using at least one *Evaluation Method*.

An Evaluation Method constitutes a paper's research design, representing a structured procedure for data collection and analysis, typically characterized by a specific name (e.g., interview). For SWA research, we identify *13* distinct Evaluation Methods. (cf. Figure 7.5) in total. The three research items (metadata) *Research Object*, *Evaluation Method*, and *Property* constitute the concept of *Findings*.

In turn, these *Findings* are supported by *Validity* and *Evidence*. Therefore, based on limitations in the method design, *Threats to Validity* should be considered and explicitly discussed when assessing how valid statements about the evaluated *Property* are. Furthermore, we document *Guidelines* referenced for both the Evaluation Method and the discussion of Threats to Validity. Additionally, each paper may be linked to external resources, typically provided via URLs, that include any kind of *Replication Artifacts*. Finally, *Metadata* is concerned with (conventional) information and data about research artifacts, such as bibliographic data for papers and identifiers or locators for replication packages.

In summary, Figure 7.3 depicts the key knowledge-based descriptive metadata elements (e.g., *Research Object*, *Property*, and *Evaluation Method*) and their interrelationships, which define the data schema used in the data extraction form of our literature study (cf. Table 7.1).

7.5. Evaluation of the Schema

In the following, we use our evaluation approach for classification schemas (cf. Section 5.3), investigating the structure's suitability (internal evaluation strategy), applicability (manual evaluation strategy), and the purpose (intrinsic evaluation strategy). As we aim to evaluate the concept of knowledge-based statements centered around the research findings, we individually regard the *Research Object*, *Evaluation Method*, and *Property* to regard and calculate the metrics in each research step. A detailed overview of each classification concept (category) with the name of the classes (e.g., Technical Experiment), the respective definitions, and (paper) examples are provided in the GitLab Wiki of our replication package. We answer the following validation question in the subsequent sections.

VQ I-3.1: How general, applicable, and suitable is the data schema for its purpose based on our evaluation approach?

7.5.1. Evaluation of the Structure's Suitability and Purpose

First, we evaluate the structure's suitability and the purpose of our (instantiated) classification schema with the internal and intrinsic evaluation strategy. For each classification dimension mentioned above, we decided to conduct the evaluation based on (i) objects under study (i.e., papers and relevant terms (M0 level)), (ii) a label dataset based on the schema (M1) considering our included papers, and (iii) related existing classification schemas (M2 level covering ACM Empirical Standards and the de facto / state-of-the-art standards for *Property*). As describing research artifacts based on knowledge statements is a complex task, that is widely unexplored, we regard the classification concept for the findings in the following as central.

Evaluating the Research Objects To identify research objects in SWA research, we manually analyzed included papers published at ICSA and ECSA by their abstracts (and if necessary also by the respective introduction or method design section) using string matching and similarity-based techniques. We constructed the classes for the research object, considering both generality (laconicity/lucidity = 1.0) and appropriateness (soundness/completeness = 1.0). Thus, we resulted with a set of 13 research object classes. This category exhibits full orthogonality, as all classes are mutually exclusive according to our derived definitions. This is reflected in an orthogonality score of 100%, with 13 distinct classes and 156 relevant fields out of 169 total entries in the orthogonality matrix, indicating no overlaps by showing mutual exclusiveness. However, as a paper might have several findings, they can be assigned in a multi-label fashion. Relevance is confirmed by the fact that each class was identified and labeled within the literature corpus of 153 analyzed papers, yielding a relevance score of 1.0. Furthermore, due to the absence of comparable classification schemas targeting SWA research objects at this level of granularity, the category achieves the highest possible novelty score (innovation = 1.0). However, significance could not be assessed due to the lack of prior taxonomies in this domain.

Evaluating the Evaluation Methods To define adequate classes for the category evaluation methods, we adopted the ACM Empirical Standards [232] (Section 3.1.1) as a foundational reference and iteratively extended the classification based on the SWA literature corpus. This process resulted in 13 distinct evaluation method classes. The category achieves full generality and appropriateness (1.0 each). The orthogonality of the evaluation methods is given in most cases. However, in mixed-methods studies, where each method is aggregated into a coherent evaluation strategy, the combination of quantitative and qualitative measures using diverse evaluation techniques may compromise the orthogonality of the defined classes. However, as research designs might vary in their originality and complexity, we do not advocate for a high orthogonality score to embrace the broad spectrum of evaluation practices. The orthogonality of each individual class for evaluation methods is given, as all classes in this category are mutually exclusive according to their definitions (i.e., no overlapping classes exist with an orthogonality score of 100%). All 13 evaluation methods were applied within the analyzed corpus (cf. Figure 7.5), indicating a high relevance. To address domain-specific needs, we introduced additional methods

not covered by the ACM standards, such as Motivating Example, Argumentation, and Verification (non-empirical), demonstrating innovation in the context of SWA evaluation.

Evaluating the Properties For the category of properties, we adopted established standards from both de facto standards and scientific literature. Therefore, we regard the classes of the categories Product Quality, Quality in Use, and Quality of Analytical Method with appropriate generality (laconicity/lucidity = 1.0) and coverage (soundness/completeness = 1.0). The orthogonality of the properties is given, as all classes in this category are mutually exclusive according to their definitions (i.e., no overlapping classes exist with an orthogonality score of 100%; 13 classes for properties, 169 total fields / 156 relevant fields in the orthogonality matrix; no overlaps and an orthogonal count of 156). Although the innovation score is zero due to the reliance on existing standards, the classification remains relevant. However, not all property classes were applied in the literature corpus: Product Quality was used in 8 of 8 classes, Quality in Use in 5 of 5, and Quality of Analytical Method in 4 of 9 (~0.44). This limited application may reflect the scope and sample of the dataset and requires further investigation to assess the practical relevance of underutilized property classes.

7.5.2. Evaluation of the Applicability

Second, we report on the manual applicability of our classification schema based on the conducted literature study, as specified in Section 7.6. Following the defined search strategy and selection criteria (cf. Section 7.6.1 and Section 7.6.2), we obtain 153 included papers for the data extraction phase. Therefore, we assign each paper to at least two reviewers. They independently classify the papers' content according to our proposed schema using our proposed documentation process (C2). Upon completion of the individual assessments, discrepancies between reviewers were systematically discussed and resolved. This process demonstrates the practical feasibility and reliability of the schema when applied manually (cf. manual evaluation strategy). In addition, we derived a gold standard dataset for our evaluations in Part III.

In addition to the manual evaluation, we examine the automated applicability of our classification and data schema, specifically, the extent to which an automated classification approach can accurately categorize paper abstracts according to the proposed schema. This aspect is addressed in Chapter 12, where we present and evaluate our framework for automated classification tasks using LLMs supported by transfer learning and prompting.

7.5.3. Summary and Findings

Based on the results of the applied evaluation approach regarding the internal, manual, and intrinsic evaluation strategy (cf. Section 5.3), we summarize the answer to our first validation question as follows: Our classification approach relies on general concepts to describe statements in papers according to their findings, underlying validity and the supporting evidence. These concepts refer to knowledge-based descriptive metadata and statements of research artifacts concerning the research findings (to advance state-of-the-art) as the central concept. To describe a knowledge statement based on the paper's findings, we

consider one research object that can be evaluated with multiple evaluation methods to investigate one or more properties. A paper may comprise one or more research objects (investigated with one or several evaluation methods and aligned properties) and, thus, include several knowledge statements. In our evaluation, we showed that we provided a new state-of-the-art and state-of-practice standard for research objects in SWA research beyond de facto standards of research topics such as provided by ACM CCS (see Section 3.1.4). This could majorly enhance the discoverability of research findings, and, thus, contribute to a faster progress and reuse in SWA research. Regarding the properties, we consulted the ISO/IEC 25010:2011[1] standard [114] and the Quality of Analytical Method according to Taverniers et al. [280] and showed its applicability in SWA research. However, not all properties were selected in our literature corpus for the respective research objects and evaluation methods. This diversity is also indicated by our second validation question. This highlights a key challenge in deriving common evaluation guidelines. A systematic approach would help to determine whether specific properties and evaluation methods are inherently suited for particular research objects, while also revealing gaps and inconsistencies in the literature; particularly in the alignment of research objects, evaluation methods, and evaluated properties. For the evaluation methods, we adopted the ACM Empirical Standards for SE research. While these standards are broadly applicable across the SE community, our study required the integration of domain-specific methods, indicating that general concepts for SE may act as a backbone for knowledge descriptions of research artifacts, but may need to be tailored to domain-specific aspects.

7.5.4. Threats to Validity

The evaluation of the research goals *structure's suitability* (internal evaluation strategy) and *purpose* (intrinsic evaluation strategy) has only been performed by the author of this thesis (and based on the discussions with the three main authors of the literature study) and is, thus, based on existing experience and research knowledge. However, the adherence to our metrics and guidelines is given (cf. Chapter 5 and Chapter 6).

External Validity refers to the generalizability of our classification and data schema and is given by the general concepts of research *Findings*, *Validity*, and *Evidence*. For each concept in this kind of knowledge statement, we derive (descriptive) metadata (categories in a taxonomy) and dedicated classes. The external validity and generalizability of our results are given, as we considered de facto (ISO 25010 (2011) [114]) and community standards (ACM SIGSOFT Empirical Standards for Software Engineering [232]) for benchmarking. To construct the research objects, we took a random set of ECSA and ICSA papers that may potentially challenge the generalizability of the findings. As we could show the completeness of the instantiated research objects in our conducted literature study, we assume easy application of the newly created classes to other SWA publications.

[1] Note: The ISO/IEC 25010:2011 [114] Standard was revised by ISO/IEC 25002:2024, ISO/IEC 25010:2023, and ISO/IEC 25019:2023, indicating the evolution of classifications and data schemas as inevitable change. This highlights the need for our general research process, as described in Chapter 8 and embedded in Figure 7.2, to understand previous concepts and maintain developed classification schemas over time.

Internal Validity of our classification schema is given by the fact that we describe knowledge statements in a paper which are inherently related and individually mutually exclusive to ensure no information loss. Each quality criterion for a dedicated evaluation goal and its corresponding metrics in the aforementioned evaluation strategy was independently compiled. The evaluation was conducted by the author of this thesis, informed by discussions with the three main authors of the literature study.

Construct Validity is addressed by applying our proposed GQM-based evaluation plan (Section 5.3). Therefore, we aligned evaluation properties and corresponding metrics to the goals and research objectives *structure's suitability*, *applicability*, and *purpose*. The applied metrics are common and established in scientific literature and/or based on newly developed evaluation concerns (e.g., relevance, and innovation for the purpose), already showing feasibility in subsequent research (see VQ I-1-2).

Confirmability and Repeatability are combined discussed as potential threats. To increase reliability and comprehensibility for other researchers, we publish all evaluation data and followed our guidelines for tool-supported data extraction and documentation, as introduced in Chapter 6. Our GitLab Wiki of the respective literature study comprises all relevant information about the process and the specification of our schema, thereby supporting independent verification and reuse.

7.6. Review Protocol

We conducted a systematic literature study following the guideline of Kitchenham and Charters [147] using our proposed classification and data schema in the data extraction phase (cf. Section 7.6.4). The guideline proposes three phases for the review process: planning, conducting, and reporting (dissemination) the literature study. In this section, we focus on the planning phase by presenting our final review protocol that contains a specification of the basic review procedures to foster replicability and a common understanding for review conduct.

7.6.1. Data Sources and Search Strategy

For our literature study, we employ a manual search strategy to collect research papers related to SWA over a time span of 5 years. To ensure the inclusion of high-quality and relevant studies, we focus on the *International Conference on Software Architecture* (ICSA) and the *European Conference on Software Architecture* (ECSA) series from 2017 to 2021. Both venues serve as primary conferences for presenting and discussing novel ideas, approaches, and scientific progress in SWA research. The selected papers are retrieved from the DBLP computer science bibliography[2].

[2] `https://dblp.uni-trier.de/db/conf/icsa/index.html` and `https://dblp.uni-trier.de/db/conf/ecsa/index.html` [Last accessed on 2024-01-29]

7.6.2. Study Selection

After obtaining SWA papers through the manual search, we apply selection criteria to assess their relevance and filter them w.r.t. our research objective. We specify the inclusion and exclusion criteria in the following.

The **inclusion criteria (IC)** of this literature study are:

IC1: Papers published at ICSA and ECSA between 2017 and 2021 (i.e., we investigate SWA papers in a time span of 5 years).

IC2: Papers assigned to at least one of the following paper classes, based on the classification by Wieringa et al. [305]: *evaluation research*, *validation research*, *proposal of solution*, and *philosophical paper*.

IC3: Papers mainly reporting primary research; as secondary and tertiary studies (i.e., literature studies of primary or secondary research data) are not in focus.

IC4: Papers published at the main proceedings of the respective conferences.

IC5: Papers with full-text available through online access.

IC6: Papers written in English.

The **exclusion criteria (EC)** of this literature study are:

EC1: Papers assigned to the paper classes *personal experience papers* and *opinion papers* (according to Wieringa et al. [305]), as they typically do not require a (strong) empirical evaluation.

EC2: Papers focused on teaching or meta-research aspects (cf. Section 2.1.1) in SWA, as these papers do not directly target research objects related to architecting activities, such as architectural decision-making and documentation.

EC3: Short papers are excluded due to their limited space for a detailed evaluation description and their typical focus on presenting work in progress (WIP) or new ideas. Consequently, these papers have lower requirements regarding their evaluation in a peer-review process. We identified short papers by the number of pages given in the calls for papers.

7.6.3. Study Quality Assessment

We aim to analyze the current state-of-practice at the ICSA and ECSA series between 2017 and 2021 and evaluate our proposed data schema. Therefore, we intentionally did not apply further quality assessment. Excluding papers via a quality assessment would bias the perception of the current state and would be a further threat to validity (cf. Section 7.6.7). Instead, we provide proposals for the SWA community based on our results and findings from the literature study to contribute to reflected research practices (cf. Section 7.9).

7.6.4. Data Extraction Form

Based on our research objectives and the corresponding RQ and VQs, we developed a data extraction form (cf. Table 7.1). While Figure 7.3 illustrates the interrelations of the aforementioned research items and extracted data, Table 7.1 provides details about the corresponding data extraction form. The complete form with the definition of its contents is available in the replication package [156], which also contains a detailed description of

each research item, such as individual *Research Objects*. To support the classification of papers using our data extraction form, we applied the documentation process, as described in Chapter 6 using the SLR Toolkit [82]. In the following, we describe each data extraction item (metadata of research artifacts) in detail.

(Descriptive) Metadata (of a Paper) Besides the extraction of bibliographic metadata of a paper (from the `.bib` file), we specify a dedicated Paper ID and classify a paper based on its research type. Therefore, we adopted the classification proposed by Wieringa et al. [305] (i.e., Evaluation Research, Proposal of Solution, Validation Research, Philosophical Paper, Personal Experience Papers) to categorize papers in requirements engineering. Notably, in accordance with Wieringa et al., a paper may be assigned to multiple classes. Identifying paper classes required a thorough understanding of the paper's content. Consequently, this information was manually extracted during the review process.

Research Object The research object of a paper refers to the central object of investigation by the researchers (authors). Authors typically explicitly state the research object (e.g., "In our paper, we present a design method…"). To ensure meaningful results, the classification of research objects must be neither too fine-grained nor too coarse-grained (cf. evaluation results of the structure in Section 7.5). We extracted and evaluated a set of 13 research objects in a pilot study using a sample of papers from both conferences that proved to be stable in the main study for ECSA and ICSA. As our research aims at the evaluation of architecture research, and not at paper reports according to Wieringa et al. [305], we investigate how specific *Research Objects* are evaluated. Notably, a single paper may present multiple research objects, as described in Section 7.4.

Evaluation Method The evaluation method refers to the approach used to assess a research object (cf. Section 7.4). Authors typically state the evaluation method explicitly (e.g., "We conducted a case study…"); however, these statements can have varying interpretations. For instance, reported case studies may refer to either large example systems or real-world cases. Given this fact, Wohlin [310] discussed the perception of misuse of the label "case study" and verified it as correct. To determine the evaluation methods, we use the ACM Empirical Standards [232] for comparison. Therefore, we map the evaluation methods used in the reviewed papers to the corresponding method description in the ACM Empirical Standards. The detected evaluation methods are listed in Figure 7.5. In some cases, certain (evaluation) methods can be employed either to construct or validate a theory. For instance, Haselböck et al. [96] use a literature review and an interview to develop decision models for microservices, which are not considered as evaluation methods in this context. In addition, we collect any evaluation guidelines cited and used by the authors.

Property Researchers evaluate specific *Properties* of *Research Objects*, which are often explicitly stated by the authors (e.g., "We evaluated the feasibility of our approach…"). Multiple properties can be evaluated with each evaluation method. In our SLR, we use a common (standardized) set of potential properties: (i) the properties defined by ISO 25010

Table 7.1.: Data extraction form of our literature study (Table extended from [153])

Data Item	Description and Options
	(Descriptive) Artifact Metadata
Bibliographic Metadata	**Descriptive Metadata of Paper**
DOI	Digital object identifier of paper (persistent identifier)
Title	Title of the paper
Authors	Authors of the paper
Venue	International Conference on Software Architecture (ICSA) or European Conference on Software Architecture (ECSA)
Additional Metadata	**Descriptive Metadata of Paper**
Paper ID	BibTex-Key (citation key) as unique identifier (i.e., according to pattern: *FirstauthorsurnameYear*)
Paper Class	Paper class according to Wieringa et al. [305]; 6 options: evaluation research, proposal of solution, validation research, philosophical papers, opinion papers, and personal experience papers
Knowledge-based Metadata	**(Descriptive) Artifact Metadata**
Findings	**Scientific Contributions**
Research Object (RO)	Investigated object(s) of research (13 options)
Evaluation Method (EM)	Applied evaluation method according to ACM Empirical Standards [232] and beyond (i.e., domain-specific methods for SWA research) (13 options)
Property	Property evaluated with a given EM for a given RO Quality in Use acc. to ISO/IEC 25010:2011 [114] (5 options) Product Quality acc. to ISO/IEC 25010:2011 [114] (8 options) Quality of Analytical Method acc. to Taverniers et al. [280] (9 options)
Evidence	**Replication Artifacts**
Tool Support	3 options: {available, used (but not available), none}
Input Data	3 options: {available, used (but not available), none}
Replication Package	2 options: {yes, no} for availability
Validity	**Validity of Findings**
Threats to Validity (TtV)	Categories of validity threats according to Feldt and Magazinius [64] 5 options: internal validity, external validity, construct validity confirmability, repeatability
Guidelines	**Methodological Guidelines (Publication Artifact)**
EM Guidelines	2 options: {yes, no} for usage separate documentation of DOI or ISBN of referenced EM guideline
TtV Guidelines	2 options: {yes, no} for usage separate documentation of DOI or ISBN of referenced TtV guideline

(2011) [114] for *Quality in Use* and *Product Quality* and (ii) *Quality Criteria for Analytical Methods*, as introduced by Taverniers et al. [280].

Replication Package We check whether a replication package (i.e., a complete collection of materials, such as datasets, source code, experimental settings, configuration files, and documentation, that enable other researchers to replicate, reproduce, or extend a study) is publicly available (i.e., *Replication Package* $\in \{yes, no\}$).

Tool Support and Input Data During the evaluation of architecture research objects, tools and/or input data are often used. As tool support, we regard tools that are not exchangeable for the evaluation, e.g., tool prototypes for an approach or a specific form for a questionnaire. Input data is the data used as input to the evaluation, e.g., models, SWA descriptions, or transcribed answers to questionnaires. We identify whether they are mentioned in the given paper (categorizing them as *used*) and whether they are (still) publicly *available*. Tool support and input data are elicited only on the level of papers, not research objects, because the relation is not always clearly described.

Threats to Validity (TtV) We classify the authors' statements about TtV of their evaluation. For comparison, we grouped TtV based on Feldt and Magazinius [64]: *Internal Validity*, *External Validity*, *Construct Validity*, *Confirmability*, and *Repeatability* (i.e., dependability in [64]). In addition, we collect the corresponding guidelines, if any were referenced.

7.6.5. Data Extraction Process

For each paper, two researchers independently extracted the data according to our predefined `.taxonomy`-file of the SLR Toolkit (Section 6.3.1), based on the specified data extraction form (Table 7.1). Papers were randomly assigned to reviewers, who classified them according to the schema and data extraction form. Any discrepancies and disagreements between their extractions were discussed pairwise; unresolved conflicts were addressed by a third reviewer (i.e., one of the main authors). To document and manage the review process, we used our guideline, as described in Section 6.4.

7.6.6. Data Synthesis

For data synthesis, we quantitatively collated and summarized the results of the included papers to analyze the relationships between the extracted data elements defined in our schema (cf. Figure 7.3). This synthesis provides an overview of the current state-of-practice in SWA research and highlights the most frequently referenced evaluation guidelines (see Table 7.2). For information visualization, we followed the supplementary materials provided by the ACM SIGSOFT Empirical Standards for systematic literature reviews [232] and developed a serverless visualization tool, *Visulite* (see Section 11.4), which enables the interactive exploration of the synthesized data.

7.6.7. Threats to Validity

We discuss potential biases based on the guideline of Kitchenham and Charters [147] in the following categories:

Selection and Publication Bias (Allocation Bias) represent a primary threat to validity, particularly w.r.t. the completeness of included papers. To mitigate this risk, we systematically conducted a manual search, i.e., without random selection, of SWA papers published at the ICSA and ECSA conference series over a five-year period (2017–2021). Consequently, the results and insights derived from our study are strictly limited to this body of literature. This limitation can be addressed in future work by extending the considered time frame, incorporating additional SWA venues, or employing additional search strategies (e.g., database searches) to validate and refine our results. We consider ICSA and ECSA as the major venues for SWA research. While ICSA and ECSA are considered as the primary venues for SWA research, other relevant venues include more general conferences (e.g., the International Conference on Software Engineering ICSE), related journals (e.g., the Journal of Software and Systems JSS), and co-located workshops at ICSA and ECSA.

Measurement and Exclusion Bias concerns bias in paper selection and data extraction. To mitigate this potential risk, we applied predefined inclusion and exclusion criteria derived from the pilot study, ensuring that only full papers within the SWA research field relevant to our research objectives were selected and further analyzed. Prior to the main study (cf. Section 7.3), we conducted a briefing with all reviewers to provide guidance for the review process and technical support. The review instructions were documented and maintained in a collaborative Git-based wiki. For the data extraction, we manually assigned two independent reviewers for each ICSA and ECSA paper. The classification schema proposed for data extraction was predefined through a pilot study and subsequent evaluation to ensure consistency. Any questions and ambiguities during the data extraction were resolved through discussions with the first three authors of our paper, ensuring methodological rigor and minimizing subjective interpretation.

Performance Bias refers to bias in the conduction of a literature study. To mitigate this potential bias, we documented the method design in a written protocol derived from the pilot study and maintained all relevant information, including inclusion and exclusion criteria, the data extraction form with class definitions, and the review process, in a Git-wiki. Prior to the main study, we conducted a briefing session with all reviewers to introduce the data extraction procedure, clarify open questions, and provide guidance on tool usage. Additionally, we provided an illustrative example for the data extraction. Our systematic review process follows the guideline established by Kitchenham and Charters [147], with some minor deviations during the conduct phase. As outlined in Section 7.6.3, we decided not to conduct a formal quality assessment of the included papers. Excluding studies based on quality criteria could introduce selection bias and foster a threat to validity. Instead of evaluating papers based on their empirical method design, our focus was on analyzing the state-of-practice in reporting research findings and their supporting evidence.

7.7. Results of the Literature Review

Following the conduct of our review protocol, applying defined inclusion and exclusion criteria after the manual search, we obtained 153 full technical papers in SWA research for further analysis using our classification schema. This section presents the relevant results to answer the validation question VQ I-3.2 (Section 7.8), which focuses on describing the state-of-practice based on our proposed and evaluated schema. A complete list of papers with the extracted raw data is available in our replication package [156, 155].

To capture the **Findings** of a paper, we extracted the research objects under investigation, the evaluation methods used, and the evaluated properties. Figure 7.4 shows the proportion of research objects over time (time span: 2017–2021). Analogously, Figure 7.5 shows the proportion of evaluation methods over time. For a more profound analysis of the **evaluation practices** in the investigated body of literature, we visualize the relations between (i) *Research Objects* over the evaluated *Properties*, and the applied *Evaluation Methods* in Figure 7.6 and (ii) *Evaluation Methods* over the evaluated *Properties*, *Threats to Validity*, and the availability of *Replication Artifacts* in Figure 7.7. While *Properties* and *Evaluation Method*s were mapped to the respective *Research Object*(s) of a paper, note that the artifact availability and *Threats to Validity* discussions were determined for each paper.

To assess artifact availability (**Evidence**) for replication, Figure 7.8 presents the proportion of papers per year categorized as follows: *packaged* (i.e., papers providing a complete replication package), *available* (i.e., papers making the employed tool or input data publicly accessible), *used* (i.e., papers utilizing unavailable tools and input data), and *none* (i.e., papers that did not employ any tool or input data).

Threats to Validity (**Validity**) are discussed in 97 papers, with external (83), internal (64), and construct validity (45) being the most common. However, reporting practices vary widely. Only 26 papers (17%) referenced evaluation guidelines, and 22 papers cited guidelines specifically addressing threats to validity, as listed in Table 7.2.

7.8. Findings and Discussion

In the following section, we use our classification and data schema (Figure 7.3) and the corresponding data extraction form (Table 7.1) to analyze and assess current state of SWA research in a time span of 5 years (2017–2021) by answering our validation question VQ I-3.2. To answer this question, we subdivide it into the following five validation questions:

VQ I-3.2.1: What is the distribution of research objects and their evaluations? How did they evolve over time?

The distribution of *Research Object*s (cf. Figure 7.4) highlights *Architecture Analysis Methods* (31 papers) and *Architecture Design Methods* (26 papers) as the most frequently investigated topics. Following these, research objects ranging from *Architecture Decision Making* (18 papers) to *Reference Architecture*s (16 papers) also appear prominently. Less frequently investigated are *Architectural Assumptions* (3 papers) and *Technical Debt* (3 papers).

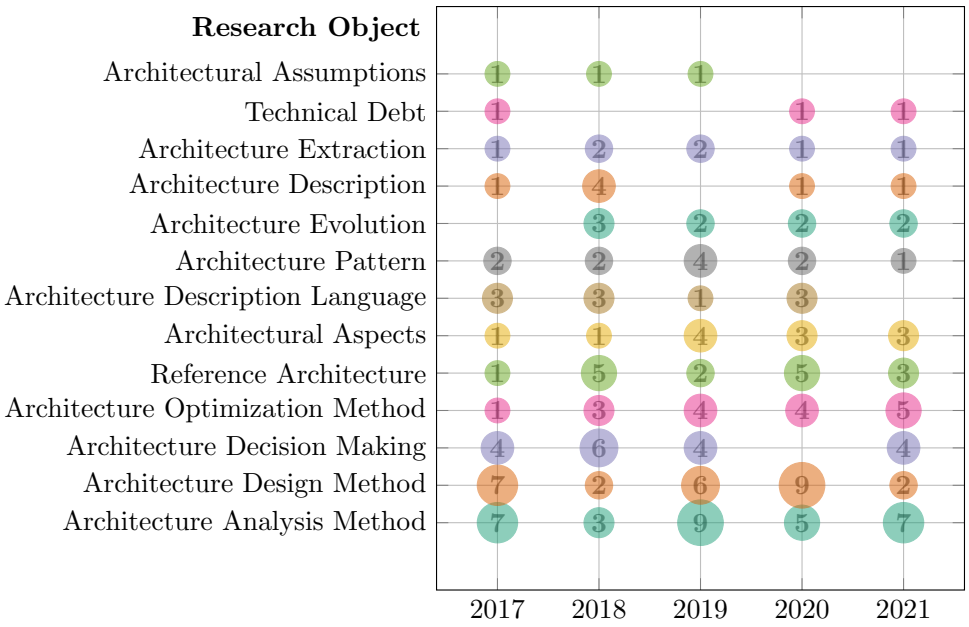

Figure 7.4.: Number of research objects from 2017 to 2021 [153].

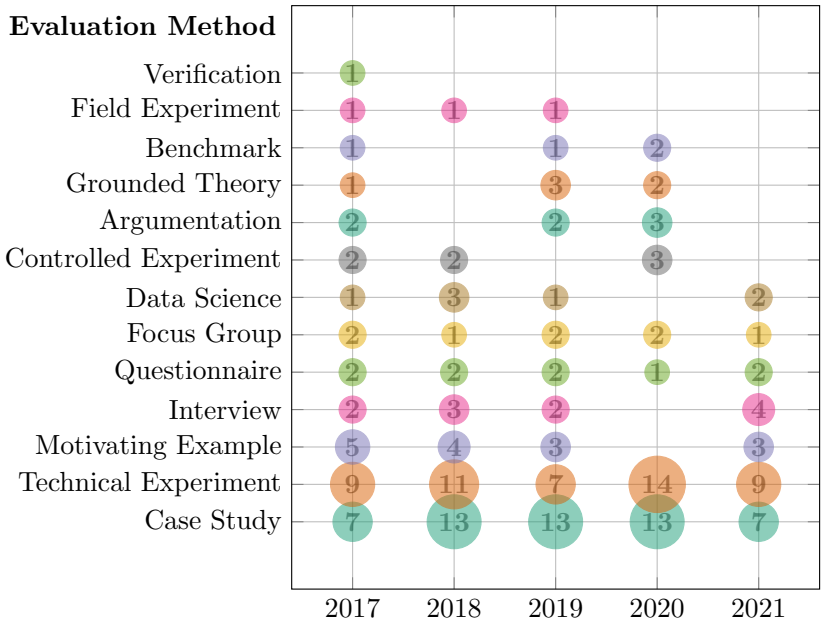

Figure 7.5.: Number of evaluation methods per research object from 2017 to 2021 [153].

Beyond these aforementioned areas, a wide range of additional research objects is addressed across the considered body of literature, indicating that SWA research presented at ICSA and ECSA is characterized by considerable thematic diversity.

The classes *Case Study* and *Technical Experiment* represent the predominant evaluation methods employed in the analyzed literature (see Figure 7.5), collectively accounting for up to 57% of all evaluation approaches (cf. Figure 7.7). While it is not possible to determine whether one method is superior to the other, given their differing methodological characteristics, it is evident that both approaches typically focus on investigating a single, well-defined phenomenon. Technical experiments are structured procedures designed to support or refute a hypothesis concerning a technological artifact. In contrast, case studies, as defined by Yin [323] and cited in the ACM SIGSOFT Empirical Standard, Ralph et al. [232] (2021), are "an empirical inquiry that investigates a contemporary phenomenon (the 'case') in depth and within its real-world context, especially when the boundaries between phenomenon and context are [unclear]." Due to their grounding in real-world settings, case studies are well-suited for generating evidence about specific *properties* of a research object. However, their depth and contextual specificity limit their suitability for generalization. Our analysis indicates that many of the applied evaluation methods do not explicitly aim to validate the generalizability of their findings. For instance, papers in the literature corpus employ a *Motivating Example* and/or *Argumentation* as their primary evaluation method. These studies typically present a problem through a simplified example and then demonstrate and reason about how their proposed approach addresses the problem within that context. Benchmarks were used in a limited number of cases, specifically for four distinct research objects and across seven evaluated properties, including *Performance Efficiency* (4 papers), *Accuracy* (2 papers), and *Effectiveness* (1 paper), as depicted in Figure 7.7.

Overall, the study reveals a wide variety of evaluation methods across the SWA literature. However, no systematic pattern could be identified in the selection of specific evaluation methods for particular types of research properties.

VQ I-3.2.2: What are the relationships between research objects, evaluated properties, and evaluation methods?

In theory, the selection of evaluation methods should be guided by the nature of the research object and the specific property under investigation. However, our analysis reveals considerable variation. For certain research objects, such as *Architecture Analysis Method*s, the diversity of employed evaluation methods is relatively limited (cf. Figure 7.6). In contrast, *Reference Architecture*s are evaluated using a broader range of approaches. The dominant evaluation methods, namely *Case Study* and *Technical Experiment*, are also the most frequently applied across several major research objects, including *Architecture Analysis Method*s, *Architecture Design Method*s, *Architecture Optimization Method*s, and *Reference Architecture*s. *Benchmark*s, as comparative evaluation techniques, are employed for four distinct research objects.

Across all research objects, *Functional Suitability* emerges as the most frequently evaluated property. This is expected, as the primary concern when introducing a solution is often whether it functions as intended ("Does it work?"). *Performance Efficiency* is the

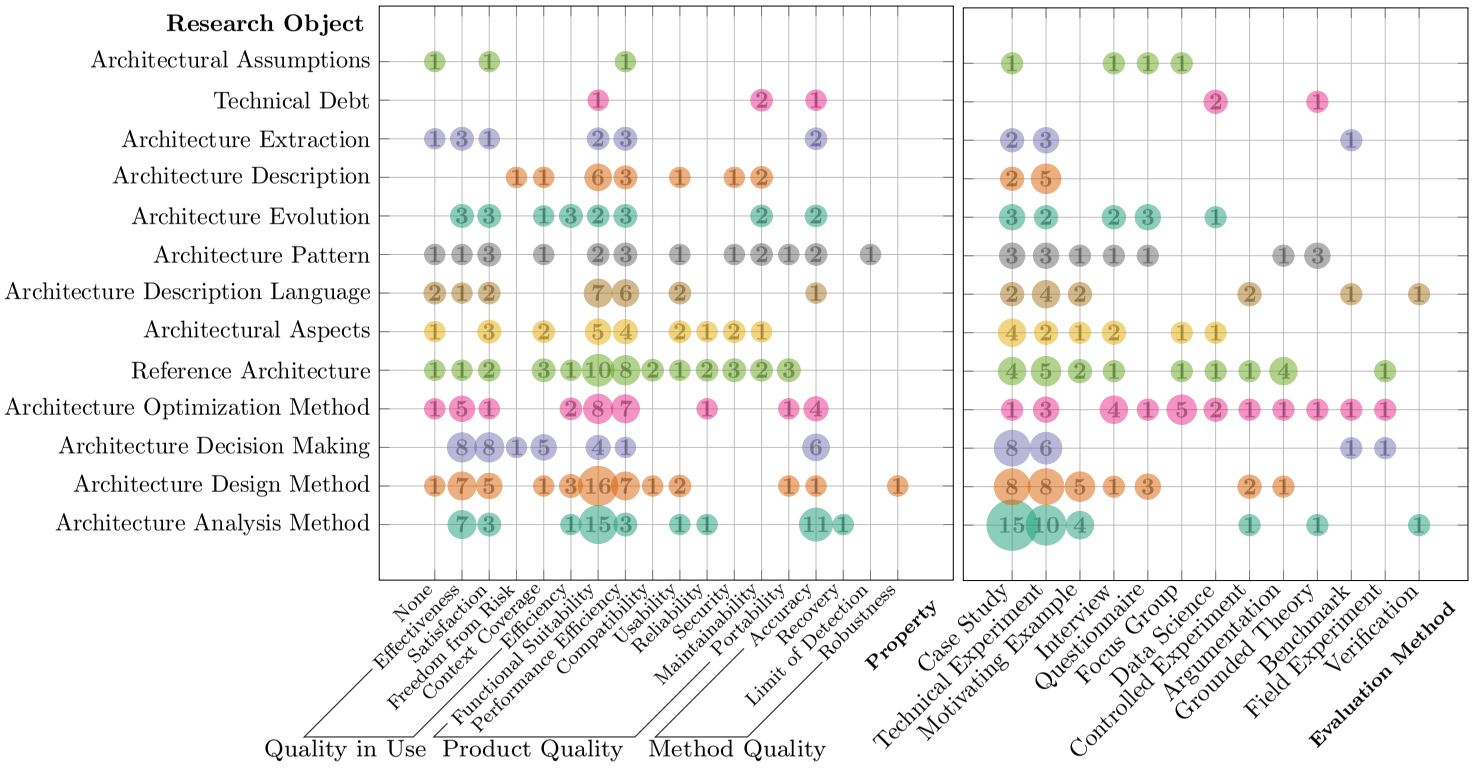

Figure 7.6.: Relations between research objects of publications, properties investigated for evaluation (a) and evaluation methods applied (b), whereas multiple evaluation methods and properties could be considered for one research object [153].

Figure 7.7.: Relations between evaluation methods (per research object) and evaluated property (a), considered TtV (per paper) (b) as well as kind of provided artifact (per paper) (c) [153].

second-most frequently evaluated property, which aligns with the computational demands of many software architecture approaches; particularly those involving pattern matching or model transformations on large-scale artifacts such as source code. Additionally, for *Architecture Analysis Method*s and *Architecture Optimization Method*s, *Accuracy* and *Effectiveness* are commonly assessed. Most research objects are associated with the evaluation of multiple properties, with *Reference Architecture* being especially diverse in this regard, as aforementioned. Conversely, research objects that are less frequently reported, tend to be evaluated with fewer properties.

No clear correspondence between evaluation methods and evaluated properties was observed (cf. Figure 7.7). *Case Study* and *Technical Experiment* are used to assess a wide range of properties. Properties categorized under *Quality in Use* are predominantly evaluated using these dominant methods, but also through approaches involving human subjects, such as *Questionnaire*, *Interview*, and *Controlled Experiment*. Properties within the *Product Quality* category are mainly evaluated using *Technical Experiment* and *Case Study*, with additional use of *Motivating Example* / *Argumentation*. In the category of *Analytical Method Quality*, *Accuracy* is the most frequently considered property.

Similar to the relationship between evaluation methods and research objects, many properties are evaluated using a variety of methods. While there is a slight and expected tendency to use human-centered evaluations for *Quality in Use* properties, the overall data does not reveal a consistent or community-wide agreement on which evaluation methods are most appropriate for specific property categories.

VQ I-3.2.3: How accessible are the evaluation research artifacts?

The distribution of evaluation artifact provision shows a trend towards increased availability of packaged artifacts (see Figure 7.8). In particular, 17 papers (11%) in 2019 and 2021 provided packaged replication artifacts. These are made available using, e.g., open access repositories or archives (Zenodo). It should be acknowledge that sharing artifacts is not feasible in all cases. The availability of replication packages may be restricted in particular cases due to non-disclosure agreements (NDAs), particularly when a study involves business-sensitive information that cannot be published. While this limitation affects replicability, these studies nonetheless contribute valuable insights into SWA research and practice. Our literature results indicate that replication artifacts are most frequently made available for *Case Study* and *Technical Experiment* (cf. Figure 7.7). This aligns with the predominance of these evaluation methods. Beyond this finding, no specific focus on particular evaluation methods was observed. Among the 50 papers employing *Case Study*, 14 papers (28%) provide access to tool support, while 17 papers (34%) make their input data available.

Table A.1 presents all the papers that provide available case studies, along with their research objects and evaluated properties. Notably, Genfer and Zdun [77] utilize their case study for two research objects, leading to the paper appearing twice in the table.

Overall, we observe that replication artifacts, especially full replication packages, are less available, facing a long-term archive problem.

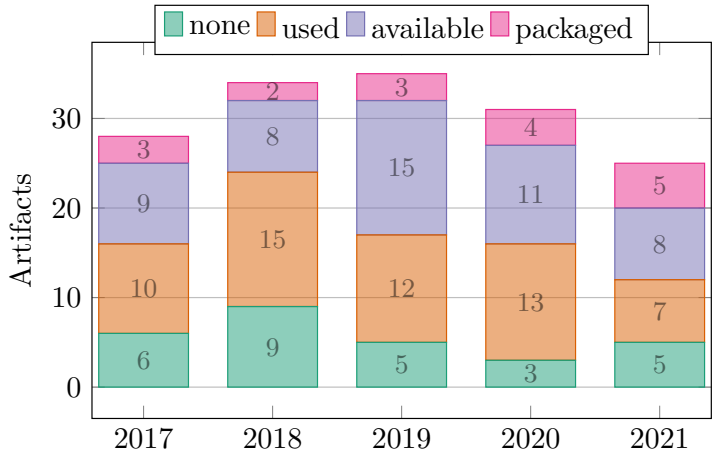

Figure 7.8.: Distribution of software architecture papers featuring a replication package (packaged), providing links to available tools or input data (available), employed tools or input data (used), or none of the above (none) from 2017 to 2021 [153].

VQ I-3.2.4: How are threats to validity discussed?

Out of the 153 analyzed papers, 97 papers (63%) explicitly address threats to validity (TtV). Among these, *External Validity* and *Internal Validity* are the most frequently discussed categories. A total of 54% of the papers consider external validity, which refers to the generalizability of the study findings. This is particularly noteworthy given that the predominant evaluation methods (among them case study) focus on the investigation of single phenomena, suggesting that generalizability is often not a central concern in the evaluation process and is frequently neither assessed nor thoroughly discussed. When examining specific evaluation methods, 17 out of the 50 papers employing case studies (34%) address external validity. In contrast, 32 of the 48 papers applying technical experiments (66%) discuss threats to external validity, indicating a higher level of consideration in this context. Only 14% of the reviewed papers (21 papers) discuss *Repeatability*. Among these, 6 of the 50 case study papers (12%) and 3 of the 48 technical experiment papers (6%) explicitly address this aspect. Additionally, studies employing questionnaires (4 out of 10) also tend to discuss repeatability. *Confirmability* is addressed in 16% of the papers (24).

Overall, this high degree of variability in the reporting of TtV underscores the need for more consistent and systematic practices in evaluating and reporting validity threats in SWA research. This also includes concrete threat types in each threat to validity category and corresponding mitigation strategies.

VQ I-3.2.5: Which guidelines are used for evaluation methods (to derive findings) and threats to validity (to derive the validity and evidence of findings)?

In the investigated body of SWA literature comprising 153 papers, ~17% (26 papers) reference guidelines for evaluation methods. The most frequently cited sources are two versions of the guidelines proposed by Runeson et al. [243, 242]. These guidelines provide methodological guidance on conducting and reporting case studies. Additionally, ~14% cite guidelines related to TtV, again primarily referencing the same publications

Table 7.2.: Referenced evaluation and threats to validity guidelines (recurring only) [153]

Refs.	1st Author	Year	Title of Evaluation Guideline
[243]	Runeson	2008	Guidelines for Conducting and Reporting Case Study Research in Software Engineering
[242]	Runeson	2012	Case Study Research in Software Engineering: Guidelines and Examples
[79]	Glaser	1967	Discovery of Grounded Theory: Strategies for Qualitative Research

Refs.	1st Author	Year	Title of Threats to Validity Guideline
[243]	Runeson	2008	Guidelines for Conducting and Reporting Case Study Research in Software Engineering
[242]	Runeson	2012	Case Study Research in Software Engineering: Guidelines and Examples
[311, 313]	Wohlin	2000, 2012	Experimentation in Software Engineering
[76]	Gasson	2004	Rigor in Grounded Theory Research: An Interpretive Perspective on Generating Theory from Qualitative Field Studies
[46]	Cook	1979	Quasi-Experimentation: Design & Analysis Issues for Field Settings

by Runeson et al. [243, 242]. In these cases, TtV considerations are embedded within broader evaluation methodology guidance. A list of recurring referenced evaluation and TtV guidelines is provided in Table 7.2. Overall, the findings indicate that evaluation and TtV guidelines are rarely and inconsistently referenced across the literature, highlighting a lack of standardization in methodological practices within the SWA community.

> ***Summary VQ I-3.2***
> *Our literature study, based on our classification schema, reveals substantial diversity in SWA research objects, evaluation methods, and investigated properties. No consistent patterns were identified that link specific properties to research objects or evaluation methods. Replication artifacts, particularly complete replication packages, are rarely available, limiting reproducibility and long-term archival. Reporting of TtV is inconsistent, with no standardized documentation of threat categories, types, or mitigation strategies. Moreover, existing guidelines on evaluation and TtV are infrequently cited, indicating a lack of methodological alignment and awareness within the field.*

7.9. Proposals for the Software Architecture Research Community

Based on the results and findings of the SLR and our specified classification schema, we define proposals for the SWA research community by answering the validation question:

VQ I-3-3: What are implications and proposals (recommendations) for the software architecture research community based on our study results?

The results and findings of our literature study reveal a broad research spectrum in SWA, characterized by diverse research objects, evaluation methods, and investigated properties. Especially, no consistent pattern emerges regarding the selection of evaluation methods or properties for specific research objects. To enhance the quality and consistency of evaluation practices, aligned with the goals of meta-research (cf. Section 2.1.1), we propose five recommendations derived from the previous discussion.

Proposal P1: Demand for Replication and Generalizability We propose to foster the generalizability of evaluation results. Our literature review identified that only a limited number of studies explicitly address TtV related to *Repeatability* and *Confirmability*. This aligns with the broader observation that replication artifacts, particularly complete replication packages, are rarely provided and available, posing challenges for reproducibility and long-term accessibility. To address this gap, future efforts should promote the replication of existing studies and the application of established evaluation approaches across varied contexts. Within the SE research community, initiatives such as artifact submission calls in main tracks or dedicated artifact evaluation tracks have emerged to support this goal (cf. 1st AE track at ESEC/FSE[3] and corresponding submission guidelines[4] in 2011). Notably, ICSA introduced an *Artifact Evaluation Track*[5] in 2021, encouraging the reproduction and replication of accepted papers to enhance generalizability. Since 2024, the artifact review process has been further supported by the NFDIxCS AE Platform[6], building upon a self-hosted instance of HotCRP[7], an open-source software system for managing conference reviews, which increased the number of artifact submissions.

Proposal P2: Provision of and Request for Replication Packages We propose that the SWA research community should actively promote the provision of replication packages, e.g., by highlighting such contributions through artifact badges (e.g., ACM Artifact Badges; see Section 3.1.3). While the availability of tools and input data is beneficial, replication packages offer additional advantages, including fixed artifact versions, reproducible results, and integrated visualizations presented in a paper (cf. Section 11.4). Ferro and Kelly [66] report positive feedback on the adoption of replication badges, reinforcing their value in enhancing scientific rigor. To support confirmability and repeatability, the community should encourage the systematic provision of replication artifacts wherever feasible (e.g., in a dedicated track; see Proposal P1). In recent years (2017–2021), we have observed a small increase in the availability of replication artifacts, a potential trend that, we believe, would be valuable to sustain. Expanding the number of studies that provide empirical evidence and make their artifacts accessible would benefit the research community and foster the vision of a sustainable RDM (see Section 3.3). Based on our review, we recommend that ICSA further integrate Open Science principles into its Call for Papers (CfP), including artifact evaluation tracks and reproducibility badges. However, it is important to recognize

[3] `https://esecfse11-aec.cs.brown.edu/` [Last accessed on 2025-02-11]
[4] `https://esecfse11-aec.cs.brown.edu/guidelines.txt` [Last accessed on 2025-02-11]
[5] `https://icsa-conferences.org/2021/call-for-papers/artifact-evaluation-track/` [Last accessed on 2025-02-11]
[6] `https://ae.nfdixcs.org/` [Last accessed on 2025-10-06]
[7] `https://hotcrp.com/` [Last accessed on 2025-10-06]

that not all studies are suited to Open Science practices, particularly those involving industry collaboration, where flexibility is essential to maintain practical relevance.

Proposal P3: Development of Benchmarks We propose the development and adoption of standardized benchmarks to enable systematic comparison of competing approaches in SWA research. Despite their potential to foster technical progress and community cohesion, as noted by prior studies [266], benchmarks are rarely applied as evaluation methods, as confirmed by our literature review. Benchmark-based comparisons are widely recognized as indicators of research maturity [97], yet no standardized benchmarks are currently applied to *Architecture Analysis Methods*, despite the feasibility of comparing different analysis methods with similar objectives using common datasets. Our findings suggest that the lack of benchmark usage is not solely due to the diversity of research analysis goals (e.g., security, performance), but also to the heterogeneity of architectural representations upon which these analyses operate. *Architecture Optimization Methods* could similarly benefit from benchmark-based evaluations, particularly for assessing optimization effectiveness, as supported by Aleti et al. [5]. However, among the reviewed literature, only one study by Cardoso et al. [38] employed a benchmark for optimization purposes, using read/write operations on the Hadoop Distributed File System to evaluate improvements in dynamic architecture reconfiguration. Beyond explicit benchmark usage, it is essential to assess the suitability of the input data used in case studies for benchmarking purposes. Researchers should systematically evaluate the potential of these data sources for integration into standardized benchmark frameworks to support reproducibility and comparability in SWA research. To support the development of benchmarks, our tool Visulite serves as a knowledge base for browsing the respective studies (see Table A.1 and Section 11.4).

Proposal P4: Provision and Development of Guidelines We propose that the SWA research community should develop dedicated evaluation guidelines that extend beyond the scope of the ACM Empirical Standards [232]. These guidelines should specify which properties are relevant for particular research objects, identify suitable evaluation methods for assessing these properties, and define corresponding TtV, including specific threat types and mitigation strategies (similar to our evaluation approach for taxonomies in Section 5.3). Such guidance would support researchers in systematically designing evaluations and assist reviewers in assessing the methodological rigor of submitted papers. While effectiveness ("Does it work?") and efficiency ("Is it fast?") are commonly evaluated, our classification and data schema reveals a broader range of properties. This may reflect a perception that verifying functionality and performance is sufficient, or it may indicate a broader absence of consensus on which properties should be evaluated for specific research objects. Similarly, there is no widely accepted standard for identifying and reporting TtV across different evaluation methods, despite the availability of multiple guidelines (cf. Petersen et al. [220]). Clear guidance, particularly for early-career researchers, is needed to determine which properties to evaluate, which methods to apply, and how to document TtV, including detailed threat types and mitigation strategies beyond generic categories such as *External Validity*. This was also indicated later by Lago et al. [168] (2024) (see Section 14.1.3). An open question remains whether general SE guidelines should be adapted

for SWA or whether domain-specific standards are required. We propose using the ACM Empirical Standards [232] as a foundational framework, while allowing justified deviations to accommodate novel research approaches that may not align with existing standards.

Proposal P5: Annotation of Research Artifacts We propose a community-driven approach to support the development of classification and data schemas, aimed at fostering a shared understanding of domain-specific metadata for research artifacts. This can be operationalized through author-led annotation of research papers, as piloted by REFSQ 2024[8]. In this dedicated conference track, authors were initially asked to provide explicit availability statements for replication packages, code repositories, and datasets. Over time, the annotation process was extended to include additional descriptive metadata. Elements of our proposed classification schema (see Section 7.4 and Table 7.1), such as *Evaluation Method* and the concept of *Paper Class*, were first referenced and applied in the context of the Open Science challenges at REFSQ 2026[9], extending the default annotation framework. Furthermore, our evaluation approach (C1 in Section 5.3) enables tracking the evolution of metadata annotations, documenting refinements and innovations as community practices evolve (see Chapter 8). This iterative, community-aligned process contributes to improved transparency, reproducibility, and metadata interoperability in empirical SE research and its subfields. Establishing such shared standards will support both researchers and reviewers in ensuring consistent and meaningful documentation of research artifacts.

7.10. Limitations

Based on the evaluation of our schema (see Section 7.5) and the findings and recommendations of our literature review (see Section 7.7, Section 7.8, and Section 7.9), we identify several limitations of our novel proposed classification and data schema, both in terms of its conceptual design and its instantiation for SWA research (see Section 7.4).

Focus on SWA Research Areas For our novel classification and data schema instantiated for SWA research, we focus on core SWA research objects and did not include SWA teaching aspects and meta-research topics (cf. exclusion criteria 2 (EC2) in Section 7.6). We expect that our defined classification concepts (cf. Figure 7.3) are still valid but differ in the concrete classes, especially for the research objects. As we cannot rely on existing classification standards in this regard and faced a high effort to set a novel state-of-the-art standard for the SWA community, we omit the aforementioned SWA research areas to reliably control the solution space. This restriction has to be addressed in future work.

Coverage of Classification Concepts Our classification approach is based on describing research artifacts. In particular, it focuses on validated knowledge statements and groups

[8] `https://2024.refsq.org/track/refsq-2024-open-science`[Last accessed on 2025-09-27]

[9] `https://2026.refsq.org/track/refsq-2026-open-science#Open-Science-Competition` [Last accessed on 2025-09-27]

them by their *Findings*, *Validity*, and *Evidence*. However, this approach may not be complete. For instance, we did not consider the research questions, the metrics aligned with the evaluation method, or the research setting for the evaluation method as further concepts for a knowledge statement. This limitation can lead to oversimplification of research works, which are inherently complex, potentially missing nuanced insights that are critical for thorough understanding, evaluation, and real-world applications (e.g., knowledge access).

Missing Multi-faceted Classification We describe gained research knowledge based on the core concepts: *Findings*, their *Validity* and supporting *Evidence*. For this, we selected a higher level of abstraction to derive classifications for those concepts. The schema operates at a descriptive categorical level, which means it may not capture finer-grained details that might be required for a comprehensive analysis. This limitation can lead to the oversimplification of complex research elements, potentially missing nuanced insights that are critical for thorough understanding and evaluation. For instance, *Threats to Validity* are described according to their categorical names, but miss a different facet to categorize their mitigation strategies. A more multi-faceted classification can support different kinds of knowledge statements.

Focus on Primary Research In our literature study, we explicitly focused on primary research and did not include papers that mainly conducted secondary research (cf. inclusion criteria (IC3) in Section 7.6). However, by excluding literature studies, we missed the corresponding labeling in the dataset, especially in mixed-method and multi-method approaches (e.g., Haselböck et al. [96] used a literature study and an interview to create decision models for microservices).

To address the aforementioned limitations associated with the compactness of our proposed approach and reduced expressiveness to other knowledge statements, we developed a general research process by unifying different classification concepts (cf. Figure 7.2). This approach is designed to support change management, particularly for maintenance purposes. The details of this approach are presented in Chapter 8.

7.11. Conclusion

This chapter addressed key limitations in describing research artifacts in SE, namely, the reliance on conventional document-based metadata, which often fails to capture the underlying knowledge embedded in artifacts such as papers, software, and data. In response, we introduced a novel classification and data schema for categorizing and describing SE research artifacts through structured, knowledge-based statements. This schema is grounded in a generalizable (and reusable) classification concept that distinguishes between a paper's *Findings*, i.e., validated contributions that advance the state-of-the-art, and the *Evidence* supporting them, assessed through their *Validity*. To show the feasibility of our approach, we instantiated this schema within the context of SWA research. We evaluated the classification schema with (i) our evaluation approach (cf. Section 5.3) and (ii) conducted a literature study of 153 full technical papers published at ICSA and ECSA between 2017

and 2021 to categorize them according to our schema. This study provided insights into current practices in knowledge representation and categorization within the SWA domain, provided an overview of the current state-of-practice in SWA research, and laid the foundation for constructing a structured knowledge base in a FAIR-compliant infrastructure (see Chapter 11). The respective and derived limitations, discussed in Section 7.10, inform the development of a general research process for refining and maintaining the classification schema (see Chapter 8). This process supports the long-term goal of improving the representation, accessibility, and reuse of research knowledge in SE.

8. A General Research Process and Maintenance Approach for Software Engineering Research Knowledge

This chapter builds on the publication "Unifying Classification Schemes for Software Engineering Meta-Research" [137] published at the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM2022). Based on the results, insights, and identified limitations in Chapter 7, and our first research goal and research objectives in Part III, we ask the following research question along with its corresponding validation question:

RQ I-4: How can classification and data schemas for knowledge description of software engineering research artifacts be maintained over time to accommodate evolving research practices and metadata requirements?

VQ I-4: What kinds of descriptive classification and data schemas in software engineering meta-research exist? What are the commonalities and differences between existing classifications in each meta-research area?

This chapter introduces a general research process for maintaining a knowledge description approach capturing research artifacts and their associated knowledge statements in Software Engineering (SE) research (Section 8.3). Building on the previously presented classification and data schema (Section 7.4), which organizes a paper's findings (validated statements) by their supported evidence and validity, we identified several limitations (cf. Section 7.10). These limitations are particularly relevant to the diversity of knowledge statement types and the evolving nature of research practices, which may necessitate future adaptations of the schema. Such changes may arise from shifts in research methodologies, community-driven developments, the emergence of new classification concepts in the literature, or normative considerations (see Section 8.2). To derive and support a common understanding within the SE community and to contribute to sustainable Research Data Management, we need classifications that are sustained, documented, and maintained in a unified way w.r.t. the various change triggers. As a consequence, we propose a general research process adapted from the evaluation approach introduced in Section 5.3, specifically tailored to address schema evolution (see Section 8.3.2). We use this general research process by analyzing existing classification concepts based on descriptive metadata from SE meta-research (see classification of Ioannidis et al. [112]), as they play a critical role in organizing and managing the rapidly expanding body of scientific knowledge. Therefore, we conduct a systematic literature study by collecting schemas that are proposed, applied,

or extended in SE meta-research. Based on the results from the literature study, we analyze the classification concepts and their knowledge-based descriptive metadata by inspecting their differences and their commonalities. We discuss their central concepts that should be adapted for our classification approach.

Consequently, this chapter constitutes the contribution C4.

8.1. Problem Statement

In the previous Chapter 7, we introduced a general classification and data schema to describe SE research artifacts. This schema captures a paper's findings in terms of their validity and evidence, represented through a feature diagram (M2 level), as depicted in Figure 8.5. This schema accounts for variability in knowledge statements with different kinds of validity and evidence, as defined in Section 7.2. Central to this approach is the notion that each finding is associated with at least one *Research Object*, and, where applicable, evaluated through one or more *Evaluation Methods* and corresponding (evaluated) *Properties* along with discussed *Threats to Validity* and linked data and software (*Evidence*). While this schema supports domain-specific knowledge extraction and classification (M1 level), we identified several limitations (see Section 7.10). These include the evolving nature of metadata schemas and the limited expressiveness of knowledge statements.

To address these challenges, we propose the need for a general research process and a maintenance approach to support the ongoing refinement of classification schemas. This process must accommodate various changes over time and unify classification concepts across multiple levels of data abstraction (M0, M1, and M2; cf. Section 5.2.2). This allows us to effectively manage the rapidly growing body of research knowledge.

In the following, we identify and discuss specific change triggers that may lead to and necessitate updates to descriptive metadata schemas, ensuring that classifications remain accurate, expressive, and adaptable to the evolving landscape of SE research.

8.2. Change Triggers for Classification Schemas in Research Contexts

Classification and data schemas capturing descriptive metadata to specify research artifacts are not static, but are subject to change as research practices evolve. They evolve in response to various internal and external triggers that reflect shifts in epistemological priorities, community practices, and normative aspects. In the following, we identify and present common change triggers for classification schemas in a scientific ecosystem. Note that the following change triggers may be incomplete.

Empirical Evidence and Descriptive Studies Classification schemas grounded in descriptive approaches often emerge from empirical analyses of large corpora, such as annotated research papers in evidence-based literature studies (see Table 6.1). As new literature review datasets become available or existing ones are re-evaluated (or updated), emerging patterns may challenge the validity, granularity, and terminological consistency of

existing classification categories. For instance, an increased interest in interdisciplinary research may necessitate the introduction of new labels that reflect hybrid methodological or epistemic orientations; especially in a cross-disciplinary research field such as SE research. One such example is the emergence of case survey studies in SE (cf. Melegati et al. [192]). Similarly, the term survey is used inconsistently in SE literature, referring either to questionnaire-based empirical studies or to narrative literature reviews, further highlighting the need for refined and context-sensitive classification schemas to bridge different SE research communities.

Annotation Practices in Conference Venues Annotation guidelines and practices adopted by conference venues (e.g., Open Science Challenges at REFSQ) can act as state-of-the-art standards, resulting in de facto standards. When annotation schemas are updated, either to reflect new research priorities or to improve inter-annotator agreement, these changes often propagate into broader classification schemas. This is particularly evident in venues that publish shared tasks or benchmark datasets.

Community Consensus and Dialogue Discourse Scientific communities periodically reassess classification schemas through workshops, position papers, and collaborative initiatives (e.g., ACM Empirical Standards [232]). Consensus-driven changes may arise from the collective recognition of outdated categories, the need for inclusivity (e.g., recognizing diverse research paradigms), or the desire to align with emerging fields and research problems (see the previous change trigger). These changes are often slow but deeply influential, especially when supported by leading scholars or institutions.

Normative Considerations Beyond descriptive adequacy, classification schemas are shaped by normative judgments about what should be classified in research artifacts and how. These considerations may stem from ethical imperatives (e.g., transparency in reporting methods), policy requirements (e.g., open science mandates), or pedagogical goals (e.g., improving research literacy). Normative shifts often lead to the introduction of new categories or the redefinition of existing ones (cf. discussion and analysis by Wohlin [310] on the definition of *Case Studies*).

Technological and Infrastructural Developments Advances in NLP (see Section 2.3), semantic annotation tools, and metadata standards can also trigger schema revisions. For instance, the ability to automatically extract and classify rhetorical structures from texts may encourage finer-grained classifications (as supported by our classification framework in Chapter 12; cf. Listing 12.1), while integration with research data repositories may necessitate alignment with metadata taxonomies and the evidence of findings.

Conclusion The evolution of classification and data schemas is a dynamic interplay between descriptive studies, community practices, normative goals, and technological advances. Understanding these change triggers is essential for designing robust, adaptable classification and data schemas that are up-to-date and support accurate representation

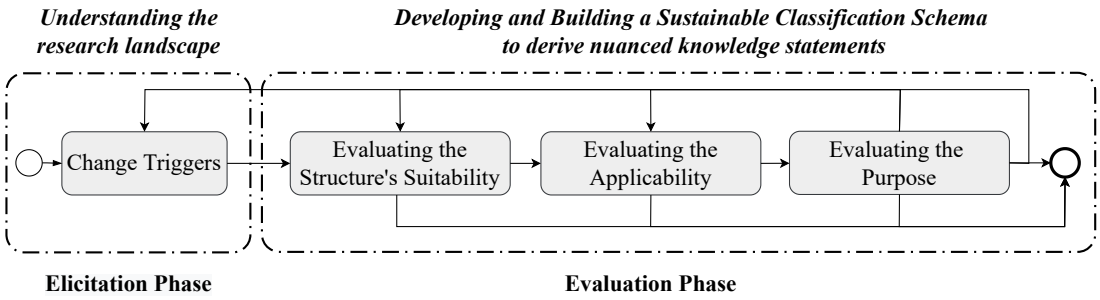

Figure 8.1.: General research process and maintenance approach

and meaningful analysis of scientific discourse to transit into machine-readable formats and machine-actionability (see the research goal of Part III).

8.3. General Research Process and Maintenance Approach

In this section, we first present our general research process (Section 8.3.1) to maintain a knowledge description approach for describing SE research artifacts and their knowledge statements and to cope with various change factors (see Section 8.2). Following, we introduce adaptations to our evaluation approach (see Chapter 5) in Section 8.3.2 that are incorporated into the general research process.

Consequently, we address our research question *RQ I-4: How can classification and data schemas for knowledge description of software engineering research artifacts be maintained over time to accommodate evolving research practices and metadata requirements?*

8.3.1. Overview of the General Research Process

Figure 8.1 depicts our general research process (i) to unify various classification concepts based on descriptive metadata to specify knowledge statements in SE research artifacts and (ii) to maintain classification and data schemas over time. This process encompasses two main phases, namely the elicitation phase and the evaluation phase. This allows us to document newly emerging categories and calculate metrics to capture and describe changes in an existing classification schema over time. The changes may arise from data at abstraction level M0, M1, or M2 (see Figure 5.3) based on the defined change trigger. To capture the complex nature of various change triggers and trigger sources, we have to adapt and extend our evaluation approach w.r.t. the metrics for calculating the generality and appropriateness of a classification and data schema (see Section 8.3.2).

In conclusion, this approach demands an iterative process to find an appropriate solution to a relevant problem and a rigorous evaluation of this solution. Our method design undergoes a series of elicitation and evaluation cycles for a stepwise improvement and refinement of the updated classification schema. This procedure terminates when insights gained from preceding evaluation activities no longer enhance the schema. The main objective is to convey how the intermediate results have been obtained.

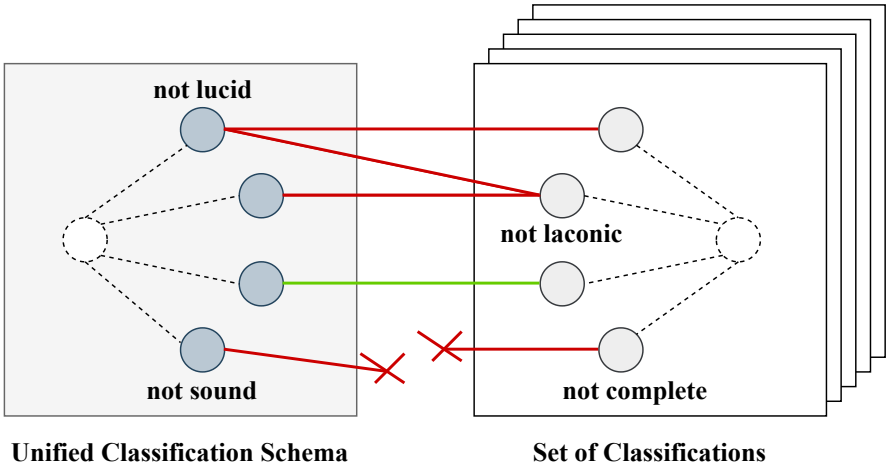

Figure 8.2.: Illustration of generality and appropriateness [137]

8.3.2. Adapted Evaluation Approach for Classification Schemas

According to Usman et al. [292] (2017), there is an increasing interest in SE classifications (cf. Section 5.1). However, they are rarely evaluated. Therefore, our general research process foresees to evaluate the structural suitability of a unified classification schema as a first evaluation step (Section 5.3). As a baseline, we check whether our classification schema is suitable to classify objects under study (e.g., digital research artifacts), i.e., we validate its generality (degree of abstraction from real-world objects) and appropriateness (degree of description coverage).

For this, our original evaluation method works on level M0 (see Section 5.2.2) to evaluate a classification and data schema. To incorporate existing or emerging classification schemas into a unified one, we have to adapt and extend the specification of our original metrics to changes on levels M1 and M2.

Figure 8.2 visualizes the concept of the metrics and provides illustrative mappings between classes of different classification schemas. In the following, we regard hierarchical classification schema structures with categories as nodes and classes as leaves.

Unification Process. To specify and formalize a unification problem, we ask the following question: Given two classes, <x> and <y>, (i.e., a syntactical label and a corresponding semantic definition) for a dedicated classification purpose, is there an abstract class <z> that is suitable for both descriptions?

This implies various operations within the hierarchical knowledge description approach for the categories (nodes including the root) and classes (leaf-nodes): insert (adding new nodes or concrete classes), remove (delete deprecated/obsolete nodes/classes), update (modify syntax and/or semantics of nodes/classes). Additionally, by unifying concepts to bridge different schemas, we support a common understanding across different SE research communities and promote sustainable literature reviews (see Section 6.2) in the long term.

Generality. The generality of a classification should be adequate by means of being both general and specific enough. This property is an indicator for a classification's granularity level and its ability to map concepts w.r.t. the classified subject matter. It determines whether a classification is too coarse-grained or too fine-grained for the intended purpose.

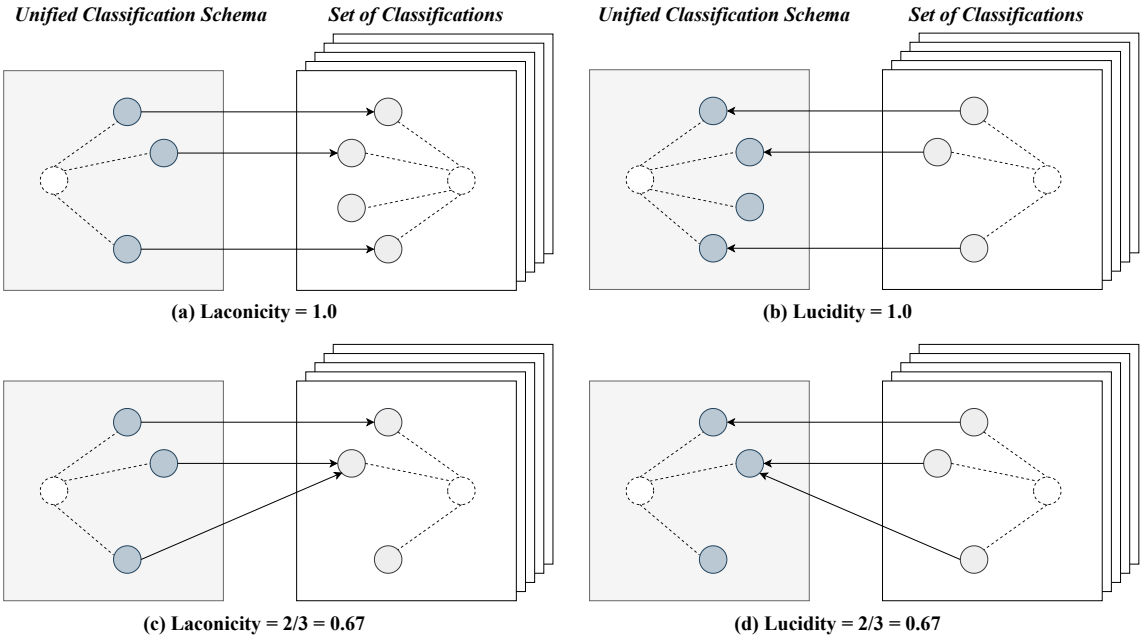

Figure 8.3.: Illustration of generality w.r.t. the metrics laconicity and lucidity

The metrics *laconicity* and *lucidity* measure the generality of a classification, i.e., whether it is both general and specific enough. Thus, this assesses the granularity of classes in a classification schema. Examples for generality and its metrics are illustrated in Figure 8.3.

Definition 8.3.1 (Preconditions)**.** Let C be a unified classification with classes $c \in C$ (i.e., leaves of a classification schema), $\mathcal{T}$ a finite set of previous classifications $T \in \mathcal{T}$ with classes $t \in T$, $m_T^C \subseteq C \times T$ a relation between classes $c \in C$ and corresponding class $t \in T$.

Definition 8.3.2 (Metrics for Generality)**.** Under the preconditions in Definition 8.3.1, a previous class $t \in T$ is *laconic* w.r.t. classification C, if there is at most one $c \in C$ with $(c,t) \in m_T^C$. The corresponding function $\text{laconic}(C,T,t)$ yields 1 if t is *laconic* and 0 otherwise.
The *laconicity* metric (i.e., conciseness) is consequently defined as:

$$\text{laconicity}(C,\mathcal{T}) = \frac{\sum_{T\in\mathcal{T}} \sum_{t\in T} \text{laconic}(C,T,t)}{\sum_{T\in\mathcal{T}} |T|} \qquad \in [0,1]$$

A class $c \in C$ is *lucid*, if there is at most one $t \in T$ with $(c,t) \in m_T^C$. Conversely, the function $\text{lucid}(C,T,c)$ yields 1 if c is *lucid* and otherwise 0.

The *lucidity* metric is defined as:

$$\text{lucidity}(C,\mathcal{T}) = \frac{\sum_{c\in C} \left(\min_{T\in\mathcal{T}} \text{lucid}(C,T,c)\right)}{|C|} \qquad \in [0,1]$$

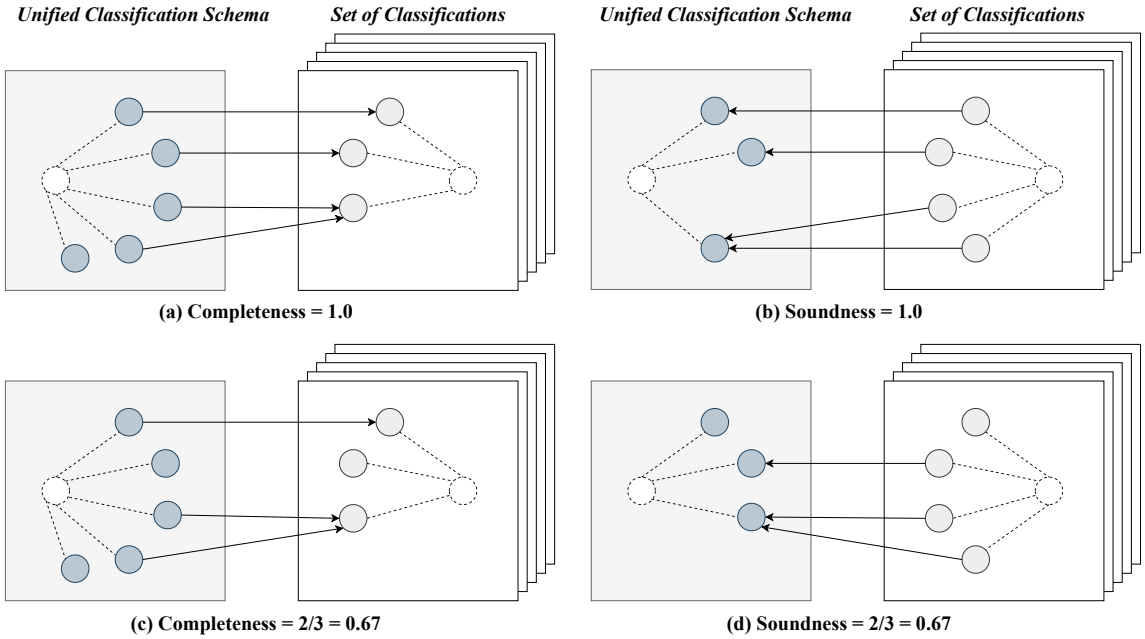

Figure 8.4.: Illustration of appropriateness w.r.t. the metrics completeness and soundness

Laconicity determines the conciseness as the fraction of laconic classes among all previous classifications, whereas a higher value is better. This holds only when already perfect classifications are unified. A low score for laconicity needs to be argued. If there are classes that are not *laconic*, there can be two main reasons. First, the unified classification might be too fine-grained, i.e., there are redundant classes in the unified classification that should be merged. Second, a previous classification $T \in \mathcal{T}$ might have classes that were split (e.g., because they were too broad and covered multiple aspects or facets). In this case, a lower score for *laconicity* can be valid.

Similarly, *lucidity* determines the fraction of lucid unified classes and categories, which should, in turn, approach 1. If a unified classification encompasses classes that are not *lucid*, the unified classification might be too coarse-grained. This means that there are unspecific classes in the unified classification that should be split up. Again, there is a second possible reason: Two or more classes of a previous classification $T \in \mathcal{T}$ were merged (due to, e.g., high semantic overlap). In this case, a lower score for *lucidity* can also be valid.

Appropriateness. The coverage of a classification should be appropriate by means of fully and correctly covering all relevant categories and classes of existing schemas. The metrics *completeness* and *soundness* measure the classification's *appropriateness*, i.e., whether it fully and correctly covers all relevant classes of previous classifications. In Figure 8.4, the metrics for appropriateness are displayed in an illustrative example.

Definition 8.3.3 (Metrics for Appropriateness)**.** Under the preconditions of Definition 8.3.1, a previous class $t \in T$ is *complete*, if there is at least one $c \in C$ with $(c, t) \in m_T^C$. The corresponding function complete(C, T, t) yields 1 if t is *complete* and 0 otherwise.
The *completeness* metric is defined as:

$$\text{completeness}(C, \mathcal{T}) = \frac{\sum_{T \in \mathcal{T}} \sum_{t \in T} \text{complete}(C, T, t)}{\sum_{T \in \mathcal{T}} |T|} \qquad \in [0, 1]$$

Likewise, a class $c \in C$ is *sound*, if there is at least one $t \in T$ with $(c, t) \in m_T^C$. The function $\text{sound}(C, T, c)$ yields 1 if c is *sound* and otherwise 0.
The *soundness* metric is defined as:

$$\text{soundness}(C, \mathcal{T}) = \frac{\sum_{c \in C} \left(\max_{T \in \mathcal{T}} \text{sound}(C, T, c) \right)}{|C|} \qquad \in [0, 1]$$

Completeness denotes the fraction of complete classes across all previous classifications. If there are previous classes that are not *complete*, the unified classification would be missing classes to cover those previous classes.

Soundness represents the fraction of sound classes in the unified classification. If the unified classification contains classes that are not *sound*, the classification would include unnecessary classes w.r.t. previous classifications.

Summary In conclusion, a unified classification to incorporate changes on different abstraction levels has a suitable granularity, if it yields a sufficiently high *laconicity* and suitably high *lucidity*. In turn, a unified classification is appropriate, if it has both sufficiently high *completeness* and *soundness*.

8.4. Research Approach and Review Protocol

In this section, we present our research approach. This includes an elicitation phase by systematically collecting classification schemas based on descriptive metadata from scientific SE literature, and identify potentials for a more fine-grained schema to enhance knowledge statements in SE research artifacts. Therefore, we introduce our review protocol in the following sections.

8.4.1. Research Aim and Questions

Based on our research question RQ I-4, we derive the need for our literature study to answer our validation question VQ I-4. Consequently, we aim to identify existing SE classifications based on descriptive metadata that can be assigned to the main meta-research areas according to Ioannidis et al. [112]. This allows us to refine knowledge statements in papers and enrich our classification schema (see Chapter 7) with new concepts.

8.4.2. Pilot Study

We conducted a pilot study in advance to formulate our final review protocol, as recommended by Kitchenham and Charters [147]. In the pilot study, we iteratively and incrementally evaluated and optimized our review protocol to identify a suitable strategy.

The goal was to refine the inclusion and exclusion criteria for the study selection process and to formulate the data extraction form in its final state. To determine a search strategy, we investigated the database search strategy. For this, we used Google Scholar (GS), which supports full-text searches, and where each search query can result in a list of up to 1,000 publications for inspection. Derived from our validation question rationale (i.e., the core elements from this question), the following search string with three search terms (concatenated with the AND operator) was defined: (i) synonyms for classifications (concatenated with the OR operator), (ii) the research domain (i.e., software engineering), and (iii) the research topic (i.e., meta-research). Since meta-research is not yet an established term and keyword in SE, the database search led to low precision and recall in terms of relevant hits and to an unmanageable number of publications for further investigation. However, we observed that several SE research classifications were (a) proposed, (b) applied, or (c) extended (updated) in secondary studies (i.e., an evidence-based research approach to conduct research on research). Consequently, we inspected secondary studies (i.e., following the evidence-based methodology) in SE per year. We included referenced classifications in these studies through a backward reference search. Furthermore, we selected a random set of publications to finalize the selection criteria and data extraction form.

8.4.3. Data Sources and Search Strategy

Based on insights from the pilot study, we formulate the following search strings per year starting with 2004 (introduction of the evidence-based research methodology in SE by Kitchenham et al. [150]) to keep the number of resulting hits under 1,000 for further data inspection:

GS search string 1 (query 1):
"systematic literature review" AND "software engineering" AND "meta analysis"
(per year from 2004 to 2020)

GS search string 2 (query 2):
"mapping study" AND "software engineering" AND "meta analysis"
(per year from 2004 to 2020)

In addition, each query per year was automatically filtered by publisher (cf. exclusion criterion EC2). By conducting this procedure, we obtained 1,255 publications in total for further study selection (cf. quantified procedure summarized in Table 8.1). Full-text reading was necessary to apply the selection criteria and to identify relevant studies.

8.4.4. Study Selection

After obtaining and collecting papers in the database search, we apply selection criteria to assess their relevance, and filter them w.r.t. our research objective. Thus, we specify the inclusion and exclusion criteria in the following.

The **inclusion criteria (IC)** of this literature study are:

Table 8.1.: Number of results by query and filtered for the database search strategy

	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	Total
Google Scholar																		
query 1	4	6	11	18	21	44	53	83	115	111	149	242	301	322	410	603	773	3266
filtered	0	1	3	6	7	13	25	28	39	36	58	90	100	99	140	209	278	1132
query 2	0	2	0	3	5	12	14	23	27	37	37	62	90	89	126	164	236	927
filtered	0	0	0	1	1	4	6	11	11	20	22	31	37	39	58	62	90	393
Merged (removing duplicates)																		
Total	0	1	3	6	8	13	26	30	44	41	61	93	113	108	158	232	318	**1255**

IC1: Papers that propose a new SE research classification approach, extend, or use a referenced classification (in a secondary study) w.r.t. the five main meta-research areas according to Ioannidis et al. [112] (i.e., Methods, Reporting, Reproducibility, Evaluation, and Incentives), whose structure is explicitly denoted as *classification*, *taxonomy*, *(classification) schema*, *classes*, *types*, or *listing*. Consequently, we regard validated schemas for classifying SE research (i.e., level M1 and M2 in Section 5.2.2).

IC2: The SE research classification is defined for general SE use and is not dedicated to a specific subdomain of SE. In this way, we can specify a classification and data schema that serves as a backbone to derive domain-specific variants.

The **exclusion criteria (EC)** of this literature study are:

EC1: Classification standards like the Software Engineering Body of Knowledge (SWEBOK)[1] or the ACM Computing Classification System (ACM CCS)[2] are excluded.

EC2: Papers where the full-text is not accessible and available online as an open manuscript from one of the four publishers: IEEE, ACM, Springer, and Elsevier.

EC3: Papers that are not written in English.

EC4: Papers that are not reported in a peer-reviewed workshop, conference, or journal (i.e., not published by IEEE, ACM, Springer, or Elsevier). Books, keynotes, posters, tutorials, extended abstracts, PhD theses, and other gray literature are excluded.

EC5: Duplicates are removed. If the same paper content is published in several venues, we consider the most recent and most complete version.

8.4.5. Study Quality Assessment

We aim to analyze existing classification schemas that capture descriptive metadata to specify SE research artifacts. Therefore, we intentionally did not apply further quality assessment, besides the exclusion criterion EC2, which eliminates papers that are not peer-reviewed. Excluding papers via a quality assessment would bias the identification of existing schemas and would constitute a further threat to validity (cf. Section 7.6.7).

[1] `https://www.computer.org/education/bodies-of-knowledge/software-engineering` [Last accessed on 2022-09-13]

[2] `https://dl.acm.org/ccs` [Last accessed on 2022-09-13]

8.4.6. Data Extraction Form and Documentation

According to Kitchenham and Charters [147], a data extraction form is necessary to accurately record the publication content and information which researchers obtain from the included studies. Therefore, we followed the documentation process for literature studies, which constituted our contribution C2. Thus, we used the SLR Toolkit [82] for the data extraction and literature data management. The tool offers four main features (cf. Chapter 6): (1) import of bibliographies and simple literature filtering, (2) design and specification of a taxonomy, (3) classification of literature, and (4) analysis of the classification with generated diagrams. In our case, we used feature (2) to extract and catalog categories and classes from the included papers in our SLR. The taxonomy specification is part of the project as *.taxonomy*-file. Consequently, for each included publication, we created a new *.taxonomy*-file. The created taxonomy can be edited in a text editor. Creating a hierarchy of extracted categories and classes is supported by the operations: (a) creation of new terms (i.e., name of a category or class), either as siblings or children of existing terms, (b) removal of existing terms as general change options, as well as (c) renaming terms to support the unification process, (d) moving a term from one super-term to another, and (e) splitting a term into multiple sub-terms as refactoring options.

Each *.taxonomy*-file begins with a preamble as comment following the pattern: Bibkey, Title, Authors, DOI, Ref. (references to other classification papers) as an optional field, and the corresponding meta-research areas according to Ioannidis et al. [112]. Table 8.2 provides an overview of our data extraction form. In addition to each extracted category and class, we provide a description or definition stated by the included papers as a comment.

In our case, the tooling worked well, as the identified classifications and their focus matched to the hierarchical extraction options of the tool. The derived datasets are machine-readable and human-understandable, and follow a tree structure. They can also be used for further data investigations and discovery.

8.4.7. Threats to Validity

We discuss threats to validity of our literature study and potential biases based on the guideline of Kitchenham and Charters [147].

Selection and Publication Bias. One of the main threats to validity is an incomplete dataset of included publications (i.e., number of identified SE research classifications). Our topic is particularly challenging, as no standardized terminology exists in SE research. The term meta-research was, at the time of this study, not an established concept in the research community nor recognized in de facto standards such as the ACM Computing Classification System (CCS). For data collection, we performed a database search by inspecting the full-text of publications that use or propose SE research classifications since 2004, when evidence-based research methodology was introduced in SE by Kitchenham et al. [150]. In addition, we performed backward reference search when a classification is used in a secondary study. To maintain the catalog of SE research classifications for the SE research community, we plan to include more scientific search engines such as ACM DL and Scopus in future work.

Table 8.2.: Data extraction form to capture classifications for the description of research artifacts in software engineering research

Data Type	Data Item	Data Description
Metadata	Reference	DOI Digital Object Identifier
	Title	Title of the selected and included publication
Collection Type	Selected	Publication that (i) proposes a new SE research classification (i.e., included), (ii) uses a former proposed classification in a secondary study (i.e., reference is included) (iii) extends (updates) a former proposed classification in a secondary study
	Reference included	Publication included by backward reference search of a selected publication
Extraction Data	Meta-research Area	The purpose of the classification in one of the main areas of meta-research according to Ioannidis et al. [112] (Methods, Reporting, Reproducibility, Evaluation, and Incentives)
	Classes	Extracted classes from included SE research classifications including their descriptions and definitions

Measurement and Exclusion Bias. We use predefined inclusion and exclusion criteria according to our research goal (i.e., identifying existing SE classifications). We evaluated them in a pilot study to avoid measurement and exclusion bias.

Performance Bias. To avoid performance bias and improve the repeatability of the SLR, we conducted a pilot study as recommended by Kitchenham and Charters [147]. We also documented our final method design in a written protocol by defining research questions, a database search strategy (including backward reference search of selected publications), filtering criteria (i.e., inclusion and exclusion criteria), and the data extraction form for further data synthesis and analysis w.r.t. the classes in SE research classifications. In addition, the data extraction and documentation process was tool-supported to obtain and follow a well-defined format.

8.5. Results of the Literature Review

In this section, we present the results from our literature study and the elicitation phase. The outcome of this phase is a set of classification schemas based on evidence-based studies that are further incorporated into the updated general schema to describe SE research artifacts. During the selection process for our literature study, several issues and contextual observations emerged that we describe in the following.

False Positives in Search Results The use of Google Scholar as a primary search engine introduced a significant number of false positives. This is primarily due to its full-text search capability, which includes keyword matches within the reference sections of publications. As a result, papers were retrieved that merely cited relevant works without themselves addressing the target concepts, thereby inflating the result set with irrelevant entries.

Cross-Disciplinary Adoption of SE Guidelines We observed that established guidelines from SE research, such as those proposed by Kitchenham and Charters [147], have been adopted in adjacent research fields, notably Information Systems (IS). While this reflects the influence and utility of SE methodologies, it also introduces ambiguity regarding domain boundaries and the contextual relevance of classification criteria for descriptive metadata concepts in SE.

Shift Toward Secondary Research Classifications Several studies included in the result set focused on categorizing secondary studies rather than primary research. This shift in classification emphasis suggests a growing interest in meta-research and synthesis in EBSE, but also highlights a gap in the direct classification of primary studies, which are essential for understanding foundational research contributions. In retrospect, we excluded such papers and focused exclusively on primary research.

Application of General-Purpose Classifications in Domain-Specific Contexts General-purpose SE classification schemas were frequently applied in domain-specific contexts. In many cases, these schemas were adapted or extended to better fit the abstraction level and specificity required by the domain. This practice underscores the flexibility of general schemas but also raises concerns about consistency and comparability across domains.

Reuse of Domain-Specific Classifications Across Domains Conversely, domain-specific classification frameworks were sometimes repurposed for broader SE contexts. For instance, the research type facets introduced by Wieringa et al. [305], originally designed for requirements engineering, were applied to classify studies in other SE subfields. This reuse necessitated adaptations to general classification guidelines, such as those by Petersen et al. [219], to accommodate broader applicability.

Limited Coverage of Non-Empirical Research in Secondary Studies Most secondary studies reviewed primary research with a full evaluation approach, focusing on empirical studies. However, papers categorized as opinion papers or experience reports are not in focus of investigations. This exclusion may lead to an underrepresentation of practitioner insights and context-rich knowledge statements for a knowledge description approach.

Literature Review Results In our literature study, we systematically collected 1,255 publications for further data selection. After applying our defined study selection criteria (i.e., inclusion and exclusion criteria), we obtained 44 selected publications for further inspection. The selected publications are composed of relevant publications that (i) proposed new classifications for SE research (i.e., new proposal for solution) that were validated via

illustrative examples or used in a secondary study or (ii) referenced classifications for data analysis in a secondary study (i.e., the referenced publications were included through a backward reference-based search). After removing duplicates and focusing on primary research, we obtained 18 included publications that classify SE research according to the five meta-research areas according to Ioannidis et al. [112] in primary research (Table 8.3). These classifications were either dedicated to one meta-research area or split their classification categories across multiple areas. Further, we had to refine the main categories to semantically group the classes for further analysis to address our first validation question and refine the expressiveness of knowledge statements in papers.

8.6. Findings and Discussion

In this section, we address our validation question VQ I-4 by analyzing the included papers from the literature study. The aim is to identify commonalities and differences in descriptive metadata types that support the formulation of nuanced knowledge statements within a structured knowledge description approach.

The included papers from the literature study were assigned to the five meta-research areas as defined by Ioannidis et al. [112] (see Table 8.2). As depicted in Table 8.3, most of the papers were categorized into *Methods*, *Reporting*, or *Reproducibility*. The remaining meta-research areas, namely *Evaluation* and *Incentives*, did not occur in the body of literature, indicating a potential gap in the current meta-research landscape within SE research.

Papers categorized under *Methods* and *Reporting* exhibit considerable heterogeneity. These works tend to propose independent classification schemas without building upon prior studies, which contradicts the principles of sustainable literature studies (see Section 6.2). In several cases, different terminologies are used to describe conceptually similar categories, or alternative facets of the same category are proposed. This lack of standardization complicates comparative analysis and integration across studies.

In contrast, papers within the *Reproducibility* category demonstrate greater consistency. Their classification schemas are either (partly) repetitive or orthogonal, allowing for the straightforward synthesis of metadata facets and indicating a strong consensus within the SE research community. This homogeneity facilitates the construction of a coherent representation of reproducibility-related metadata in SE research. In particular, these papers refine our *Evidence* concept.

To gain a more nuanced understanding of the different classification and data schemas in SE literature, we extracted and analyzed each category and class of the included papers. We distinguish between two overarching types of metadata: conventional descriptive metadata and knowledge-based descriptive metadata. Conventional descriptive metadata refers to classification concepts that characterize the research artifact (i.e., the paper itself) without directly influencing its *Findings*. For the final structuring, we only considered recurrent classification concepts / categories, as they indicate to provide stronger community acceptance and agreement.

The conventional descriptive metadata includes, besides bibliographic metadata, the (i) *Paper Class* (Wieringa et al. [305]) / *Research Facet* (Petersen et al. [221]) comprising proposal for solution, validation research, etc., and (ii) *Philosophical stances* [58] (positivism,

Table 8.3.: Overview of included publications mapped to the main meta-research areas

Ref.	Year	**Paper Title**	**Type**
Glass et al. [80]	2002	Research in software engineering: an analysis of the literature	Methods
Shaw [261]	2003	Writing good software engineering research papers	Methods
Wieringa et al. [305]	2005	Requirements engineering paper classification and evaluation criteria: a proposal and a discussion	Methods, Reporting
Zannier et al. [327]	2006	On the success of empirical studies in the international conference on software engineering	Methods
Höfer and Tichy [101]	2007	Status of Empirical Research in Software Engineering	Methods
Easterbrook et al. [58]	2008	Selecting Empirical Methods for Software Engineering Research	Methods, Reporting
Montesi and Lago [196]	2008	Software engineering article types: An analysis of the literature	Methods, Reporting
Petersen et al. [219]	2008	Systematic mapping studies in software engineering	Methods, Reporting
Cruzes and Dybå [49]	2010	Synthesizing evidence in software engineering research	Methods
Silva et al. [265]	2014	Replication of empirical studies in software engineering research: a systematic mapping study	Reproducibility
Gómez et al. [81]	2014	Understanding replication of experiments in software engineering: A classification	Reproducibility
Magalhães et al. [188]	2015	Investigations about replication of empirical studies in software engineering: A systematic mapping study	Reproducibility
Petersen et al. [221]	2015	Guidelines for conducting systematic mapping studies in software engineering: An update	Methods
Huang et al. [105]	2018	Synthesizing qualitative research in software engineering: A critical review	Methods
Santos et al. [250]	2018	Comparing techniques for aggregating interrelated replications in software engineering	Reproducibility
Stol et al. [273]	2018	The ABC of software engineering research	Methods
Shepperd et al. [263]	2018	The role and value of replication in empirical software engineering results	Reproducibility
Santos et al. [249]	2020	Analyzing families of experiments in SE: a systematic mapping study	Reproducibility

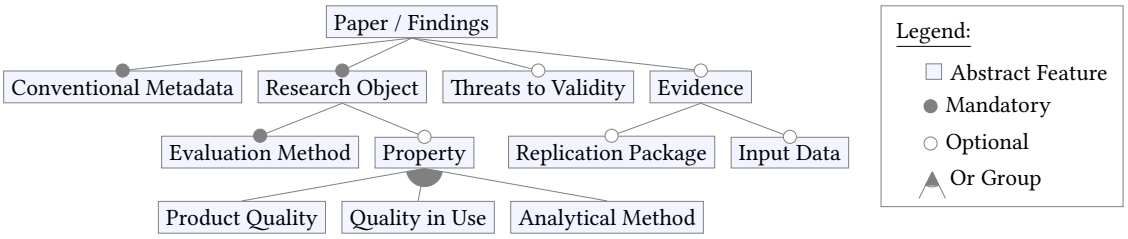

Figure 8.5.: Feature diagram of our general classification and data schema for software engineering research based on the concepts *Findings*, *Validity*, and *Evidence*

constructivism, critical theory, pragmatism). Although these classes were originally proposed for the research field requirements engineering, they became accepted in the broader SE research community (see Section 8.5). We propose a further class, namely *Replication Study* to align with the new elaborated classes for *Evidence* (see below).

The conventional descriptive metadata types were accompanied by *knowledge-based descriptive metadata*, which is our main focus for knowledge statement descriptions. Our classification and data schema comprises the *Research Object*, *Property*, *Evaluation Method*, *Threats to Validity* (see also Easterbrook et al. [58], Petersen et al. [221]), and *Evidence* categories (see Figure 8.5).

In addition, the literature reveals further relevant categories that enrich the knowledge-based schema:

- ***Research Question***, including its *Type* (e.g., exploratory, design-oriented) and *Objective* (e.g., feasibility, generalization), as discussed by Easterbrook et al. [58] and Shaw [261];
- ***Contributions***, including their *Name* (e.g., guidelines, tools, approaches, methods) and their *Type* (e.g., descriptive, evaluative, formulative/constructive, etc.), as identified by Glass et al. [80], [261], Petersen et al. [219, 221], Stol et al. [273];
- ***Research Methods***, including their *Settings* [273]; their *Type* (e.g., empirical vs. non-empirical; collection vs. synthesis), as discussed in Glass et al. [80], Shaw [261], Montesi et al. [196], Cruzes et al. [49], Huang et al. [105]; and their *Names*, see Glass et al. [80], Höfer and Tichy [101], Easterbrook et al. [58], Montesi and Lago [196], Cruzes and Dybå [49], Stol et al. [273].

These categories are particularly relevant to the meta-research areas *Methods* and *Reporting*, where we observe some overlap / commonalities and conceptual alignment. However, upon closer examination of frequently referenced categories such as *Research Method Name*, we find a lack of consistency both in the methods covered and in the terminology used. This inconsistency is also reported by Stol et al. [273], highlighting the absence of a standardized vocabulary in the field. Moreover, some core concepts and discussions are missing from the body of literature, such as applied metrics in SE research. Following the meta-research area *Reproducibility*, we incorporate additional categories into our *Evidence* concept. This concept encompasses (i) Software artifacts and (ii) Empirical artifacts (data and their facets, e.g., Source of Data [101]) as supplementary artifacts of a paper supporting the *Findings*. Existing literature provides characteristics for *Replication Studies* including (i) *Responsibility* (i.e., who is leading the replication study), (ii) the *Type of*

Procedure (i.e., how the study is replicated), and (iii) *Replication Result*, which characterizes the *Type of Results* (i.e., whether they confirm or reject the results of a prior study, or may in some cases remain undecided) and the *Aggregation Method* of results (see Silva et al. [265], Gómez et al. [81], Santos et al. [249], Magalhães et al. [188], Shepperd et al. [263]).

In conclusion, integrating these recurrent and commonly identified descriptive metadata categories into our general classification and data schema has the potential to enhance the expressiveness of knowledge descriptions, and, consequently, enable nuanced modeling of knowledge statements and support more advanced knowledge discovery.

8.7. Conclusion

In this chapter, we addressed the issue and the dynamic nature of classification and data schemas, emphasizing that such structures are subject to change over time and therefore require systematic maintenance. To this end, we first elaborate on potential change triggers within the scientific ecosystem that may influence both descriptive metadata and the foundational concept of classification schemas. Building on these considerations, and acknowledging changes at the different levels of data and knowledge abstractions (see M0–M3 levels in Section 5.2.2), we propose a general research process and maintenance strategy. This approach is grounded in our evaluation approach (see Section 5.3), with particular emphasis on adapting the metrics from evaluation step 1 to address changes at the M2 level. Specifically, we refine the internal evaluation metrics to support the consistent and sustainable description of SE research artifacts, including papers, datasets, and software. Our overarching general research approach comprises three phases, aligned with the evaluation steps and research objectives defined in our contribution C1. To demonstrate the feasibility of our contribution C4 and to refine the specified core knowledge statements, namely *Findings*, *Validity*, and *Evidence*, in response to the limitations discussed in Section 7.10, we conducted a literature study by analyzing 1,255 papers. We successively identified descriptive classification concepts from evidence-based literature and incorporated them into our schema to derive more nuanced knowledge statements. As a result, our proposed solution enables the management of various types of schema changes, particularly those occurring at the M0 and M1 abstraction levels. In conclusion, we present a general research process and maintenance strategy designed to keep classification and data schemas for SE research artifacts up to date, enable nuanced machine-readability and machine-actionability, as addressed in Part III.

Part III.

Knowledge Access in Software Engineering Research

9. Overview

Part III mainly presents the core technical contributions of this doctoral thesis, building upon the conceptual approaches established in Part II. Therefore, we address our second research goal, as outlined in Section 1.2. Each of the contributions in this part addresses a dedicated research question (cf. Section 1.4) along with corresponding validation questions.

In Chapter 10, we discuss the design rationale of the knowledge access module within our approach KARAGEN. Both modules of KARAGEN, i.e., the knowledge description module and the knowledge access module, aim at coping with evolving schemas in the knowledge description approach and in the populated research knowledge graph serving as a knowledge base. We continue to elaborate on considerations about quality assurance and evaluation preliminaries for our experimental settings in Chapter 12 and Chapter 13.

In Chapter 11, we transform our classification and data schema (for software architecture research; see Section 7.4) into an ORKG template (C5). We design the template in such a way that it covers state-of-practice and normative aspects. This template serves as a blueprint for the software architecture research community, enabling the structured and lossless representation of modular research units (i.e., a research step in a research project). In addition to this main template, we derive three further template variants (see Figure A.1) to evaluate the robustness of the retrieval approach. The chapter also introduces *Visulite*, a reusable, extensible, and serverless web application that provides visualizations of the dataset in our replication package [155] and supports dataset exploration.

In Chapter 12, we introduce our classification framework (C6) to semi-automate the task of research artifact categorization and knowledge graph population to build a knowledge base for our retrieval approach (see Section 13.2). The framework mainly supports two modern classification paradigms in AI, namely transfer learning and prompting, with various features that can be differently combined to foster robustness of classifiers. Additionally, the framework can be easily reused to derive classifiers when the classification and data schema evolves. Due to the obtained evaluation results, we also analyze the quality of abstracts in our dataset retrospectively based on our specification of structured abstracts comprising and integrating parts of Open Science and FAIR principles. This chapter closes the contributions to the knowledge description module within KARAGEN.

In Chapter 13, we lastly present our schema-agnostic retrieval approach HubLink (C7), embedded in a graph-based RAG-inspired pipeline within our retrieval framework, which operationalizes the knowledge access module of KARAGEN, enabling efficient and context-aware retrieval of scholarly knowledge.

For a research overview of Part III, please refer to Figure 1.3.

10. KARAGEN: Knowledge Augmented Retrieval and Generation Engine

In this chapter, we present the main technical contributions and outline our overall approach. This approach aims to support (i) **knowledge description** based on a systematic evaluation approach to cope with and track changes over time and (ii) scientific **knowledge access** using an `RDF`-based KG and LLMs by providing a schema-agnostic, training-free retrieval approach. We call our underlying approach KARAGEN (Knowledge Augmented Retrieval And Generation ENgine). In this context, we build upon the conceptual approaches (C1–C4) presented in Part II, while focusing on our technical contributions. Thus, we ask the following main design research questions, central for this doctoral thesis:

RQ': How to construct an approach that combines the advantages of knowledge graphs and large language models to ease (i) knowledge description and (ii) knowledge access (in software architecture research)? (cf. Section 1.3 and Section 10.1)

Our approach derived from RQ' is embedded in the architecture using RAG, as depicted in Figure 10.1. In turn, we derive this technical question from the following main question, central in our research overview, as explained in Section 1.4.

RQ: How to map contribution types and (validated) statements to template (research) questions? (cf. Section 10.3)

In the following sections, we present the design rationale of our approach KARAGEN (Section 10.1). This is followed by a set of best practices and guidelines (Section 10.2) aimed at ensuring the quality of the conducted experiments in Chapter 12 and Chapter 13. Furthermore, we introduce our question-answering taxonomy for retrieval systems (Section 10.3), which serves as the foundation for addressing our central research question RQ. Then, we present experiment preliminaries (Section 10.4) by describing our experimental environment and analyzing our gold standard dataset derived from the literature study in Chapter 7 (used in Chapter 12) and the constructed QA datasets based on our taxonomy mentioned above (used in Chapter 13). Finally, we conclude in Section 10.5.

10.1. Approach and Problem Statement

In this section, we present our idea of combining KGs and LLMs to enhance and ease scientific knowledge description and access, with a particular focus on software architecture research, for researchers and practitioners, and answer our main research question RQ'.

There are two main parts of the approach: (I) Knowledge graph population by *managing knowledge description*, and (II) *knowledge access and retrieval* by using RKGs and LLMs. The approach KARAGEN is depicted in Figure 10.1.

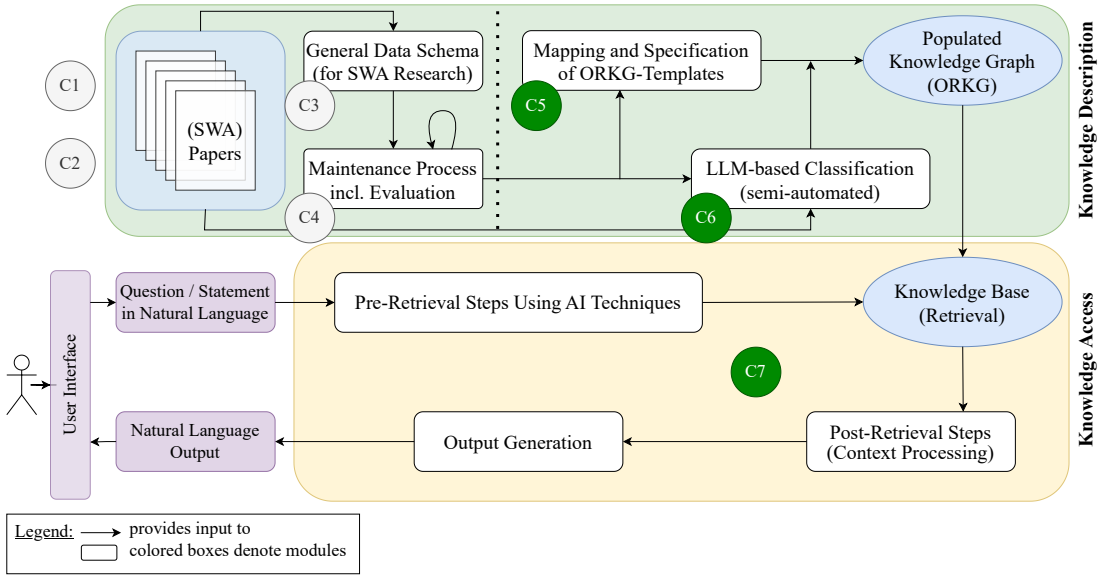

Figure 10.1.: Overview of the KARAGEN approach (Knowledge Augmented Retrieval And Generation ENgine), highlighting the contributions C5–C7 of Part III (Figure extended from Kaplan et al. [133])

Design Rationale In scientific domains, information retrieval tasks demand high recall, contextual depth, semantic interpretability, and provenance tracking for literature sources and references. While traditional Generative AI models and naive RAG systems based on vector stores and embedding-based retrieval (cf. ORKG Ask in Section 14.2.4, Consensus, Elicit, and SciSpace) offer general-purpose capabilities, they often fall short when handling the complexity and interdependencies inherent in scientific knowledge, especially within domain-specific research fields. By contrast, graph-based RAG integrates RAG with structured KG reasoning. This approach is especially well suited for scholarly and technical information retrieval, as it mitigates hallucinations, strengthens multi-hop reasoning and inference, and enhances interpretability through explicit graph-based paths. This enables us to combine and benefit from the complementary pros and cons of KGs and LLMs, as outlined by Pan et al. [213]. The main modules in KARAGEN (see Figure 10.1) are designed such that they can manage evolving knowledge description approaches. They support automated classification for knowledge graph population and enable knowledge access by schema-agnostic, training-free retrieval. This architecture and the corresponding contributions ensure maintainability of knowledge description and graph population while facilitating flexible, context-aware querying within a graph-based RAG-inspired pipeline.

10.2. Quality Assurance for Experiments

This section builds on the publication “Responsible and Sustainable AI: Considering Energy Consumption in Automated Text Classification Evaluation Tasks” [132] pub-

lished at the 9th IEEE/ACM International Workshop on Green and Sustainable Software (GREENS@ICSE2025).

As defined in our evaluation method in Chapter 5, the evaluation of the *applicability* constitutes the second step of the evaluation process. In this step, researchers assess how applicable the (hierarchical) knowledge description approach is, either through manual evaluation by humans or through automated evaluation by machines. Building upon the third evaluation step, we have to build a knowledge base that serves a dedicated *purpose* in terms of applications and functions (e.g., retrieval or metadata cataloging [237]). In the following sections, we present best practices when evaluating the *applicability* with an automated approach and the *purpose* with a retrieval approach. These best practices aim to complement concrete guidelines for empirical studies in SE involving LLMs and contribute to the meta-research area "methods", according to Ioannidis et al. [112].

Therefore, we first present the usefulness of and need for experiment tracking tools in AI-based studies and software systems (Section 10.2.1). Second, we provide foundational building blocks to track energy consumption for the estimation of carbon emissions (Section 10.2.2) in experimental setups during different phases (e.g., training, inference). Finally, we conclude with best practices for experiment settings (Section 10.2.3), especially when tracking energy consumption and considering performance metrics and sustainability aspects in trade-off (see Section 5.3.2.2).

10.2.1. Experiment Management Tracking Tools

The rapid evolution of research in AI and ML, together with the growing range of their applications in science, engineering, and practice, has necessitated the development of systematic approaches to manage and monitor experiments. These tools support researchers and practitioners in evaluating and building AI software systems [108]. They are commonly available as cloud-based platforms, such as Microsoft Azure ML[1], or as standalone software frameworks, such as MLflow[2]. In contrast to traditional SE, where assets can be managed via version control systems (e.g., Git) to effectively manage source code and related artifacts, the management of ML systems presents a substantially more complex challenge. This complexity arises from the multifaceted nature of ML assets, which include, according to Idowu et al. [108]:

- Resource artifacts such as datasets,
- Software artifacts including source code and hyperparameters, and
- Metadata, which captures experiment configurations, execution logs, performance metrics, and provenance data.

To address the intrinsic challenges of ML lifecycle management, experiment tracking tools have to support a wide range of requirements, including versioning, traceability, auditability, explainability, interpretability, collaboration, and reproducibility. These features

[1] `https://azure.microsoft.com/en-us/products/machine-learning/` [Last accessed on 2025-09-25]
[2] `https://mlflow.org/` [Last accessed on 2025-09-25]

are essential to ensure the reliability, transparency, and accountability of ML experiments. Furthermore, efficient mechanisms for querying and retrieving the results of past experiment runs are critical to facilitate iterative model development and comparative evaluation [108]. An overview of the best experiment tracking tools in 2024 is provided by Kluge and Jenkner [151]. However, scientific literature indicates that some factors affect the tools' maturity, such as the lack of interoperability across different tools (i.e., bridging the gap between development and production environments) and the lack of explicit representation of domain knowledge [37]. In the context of our research, additional considerations arise, particularly w.r.t. the monitoring and management of energy consumption and carbon emissions (cf. Section 10.2.2) during language model training and/or inference. These sustainability-related aspects are becoming increasingly important in the assessment of AI systems. Therefore, selecting an appropriate experiment management tool requires careful evaluation to ensure alignment with both technical and environmental research objectives, as specified in our evaluation as the automated evaluation strategy to assess the research goal *applicability* (cf. Section 5.3).

10.2.2. Tracking Energy Consumption and Carbon Emissions

In addition to directly measuring energy consumption from hardware components, various tools have been developed to estimate the energy usage and corresponding carbon emissions associated with AI and ML approaches. These tools provide valuable insights into the environmental impact of computational costs, particularly in NLP. In the following, we present a concise overview of selected tools, including Python libraries and browser-based applications, focusing specifically on those that are freely available and documented in peer-reviewed publications.

Carbontracker[3] (2020, MIT License) is an open-source tool designed to monitor and predict the energy consumption and carbon footprint of DL model training. Beyond retrospective reporting, the tool implements proactive prediction capabilities, enabling practitioners to reduce carbon emissions by adapting training strategies in advance. The library supports a wide range of execution environments, including high-performance computing clusters, desktop workstations, and Google Colab notebooks [314].

CodeCarbon[4] (2020, MIT License) is a Python library developed as the successor to ML CO_2 Impact [167], offering fine-grained estimation and tracking of carbon emissions. The tool accounts for a range of variables, including the computing infrastructure, geographical location, hardware configuration, resource usage, and runtime [47]. This comprehensive approach allows for improved accuracy in the quantification of emissions.

Cumulator[5] (2021, MIT License) is a Python-based package for estimating the energy consumption of computational tasks. It incorporates runtime, GPU utilization, and regional carbon intensity, while employing a fixed reference value for GPU power consumption representative of typical use in academic and healthcare contexts [289].

[3] `https://github.com/saintslab/carbontracker` [Last accessed on 2025-01-29]
[4] `https://github.com/mlco2/codecarbon` [Last accessed on 2025-01-29]
[5] `https://github.com/EPFLiGHT/cumulator` [Last accessed on 2025-01-29]

eco2AI[6] (2022, Apache 2.0 License) is a Python library that directly monitors system processes associated with model training. This process-level granularity reduces the risk of overestimation common in coarser measurement approaches, thereby improving the reliability of energy consumption and carbon footprint estimates [34].

Green Algorithms[7] (2021, CC-BY-4.0 License) is a browser-based tool that calculates energy usage and carbon emissions based on user-provided parameters. These include runtime, processor specifications (e.g., number and types of cores), memory usage, platform type (e.g., personal computer, local server, cloud computing), and geographical location [171]. By requiring explicit input parameters, the tool allows users to model diverse computational scenarios.

LLMCarbon[8] (2024, Apache License 2.0) is an emerging end-to-end framework for modeling the carbon footprint of LLM workflows, encompassing training, inference, experimentation, and storage processes. The tool is designed to facilitate design-space exploration, enabling practitioners to evaluate trade-offs between environmental impact and model performance metrics such as test loss [60].

Collectively, these tools represent a growing ecosystem of open-source solutions for quantifying and mitigating the environmental impacts of ML. Their availability under permissive licenses, such as MIT, Apache 2.0, and CC-BY-4.0, ensures accessibility for researchers and practitioners aiming to implement sustainable practices in computational modeling following the concept of Green AI (cf. Section 2.7).

10.2.3. Best Practices for Experiment Settings

Evaluating energy consumption and carbon emissions in experimental settings requires a certain care. As with other performance measurements, it is essential to ensure consistent conditions and experimental settings across all approaches (i.e., the objects under study) to enable valid comparisons. Specifically, the investigator has to perform the same classification task while maintaining stable parameters throughout the experimental runs. This consistency allows for reliable tracking of energy and emission metrics across different configurations, facilitating the identification of automated classification approaches that balance performance with environmental impact. To support this process, the following section outlines a set of best practices for investigators [132].

Use the same hardware and platform To achieve comparable and reliable results, it is important to use identical hardware configurations and the same platform (e.g., operating system, etc.). Alternative hardware or platforms may exhibit different performance characteristics, particularly in terms of energy consumption. More efficient hardware can reduce energy consumption significantly, but we are interested in measuring and comparing our software. This also implies that changing programming languages complicates the comparison, as different programming languages have different efficiencies, as Pereira et al. [217] showed.

[6] `https://github.com/sb-ai-lab/Eco2AI` [Last accessed on 2025-01-29]
[7] `https://github.com/GreenAlgorithms/green-algorithms-tool` [Last accessed on 2025-01-29]
[8] `https://github.com/SotaroKaneda/MLCarbon` [Last accessed on 2025-09-17]

Resources should focus only on the task The hardware should not be concerned with other tasks, i.e., it should remain idle except for executing the classifier under investigation. Otherwise, background activity may introduce *noise*, potentially affecting the measurements and rendering the results unreliable.

Perform the experiments at roughly the same time To minimize the influence of external variables such as fluctuations in the local energy mix or system-level changes (e.g., platform updates), it is recommended to conduct all experiments approximately at the same time. Temporal consistency helps ensure the reliability and comparability of energy consumption and emission measurements across different experiment runs.

Use the same task and evaluation setting To enable a valid comparison between multiple approaches, investigators need to evaluate the same task using an identical evaluation setting and the same dataset. Although this may sound self-evident, critical details are often overlooked. To illustrate this, we take an automated text classification task as an example. For fine-tuning, this includes consistent use of dataset splits, e.g., for training and evaluation data to avoid having a split that requires slightly more energy due to, e.g., more contained text. If investigators do k-fold cross-validation, the splits also need to be the same, preferably in the same order to avoid fluctuations.

Use the same tooling Different tooling may influence the evaluation due to varying margins of error or the overhead these tools introduce for logging or capturing energy consumption, etc. Additionally, different programming languages also typically require different tooling, which is another reason to select only one programming language.

Measure and distinguish the different phases A machine learning-based classifier typically involves at least two phases: training and inference. The energy consumption for training might differ a lot compared to the inference. For instance, some approaches might perform better in training but worse in inference. While training is often a one-time effort (except in the case of retraining), the inference is constantly needed when deploying the approach. Therefore, it is recommended to measure and report energy consumption for both phases to enable meaningful comparisons and provide a robust basis for decision-making.

Repeat experiments We recommend conducting multiple runs to minimize the impact of variance. This practice also enhances the stability and statistical significance of the measurements.

Reporting the Results Written text in papers and appropriate visualization techniques are useful tools, to communicate and interpret research results obtained from applied metrics. We recommend using tables to list the resulting values or grid reference systems to better contrast both aspects (i.e., performance and carbon emissions by, e.g., illustrating results w.r.t. relative carbon emission (CE_{rel}). We present our combined sustainability metrics in Section 5.3.2.2.

10.3. Question-Answering Taxonomy for Retrieval Systems

For the construction of a QA dataset, we first need competency questions. We constructed the competency questions based on two orthogonal characteristics: *Question Complexity* (i.e., affecting the complexity degree of the retrieval) and *Use Cases* (i.e., different combined metadata types in the input (question) and output (answer) [134]). This two-dimensional taxonomy helps to define competency questions systematically and equally distributed, preventing bias in the performance results to assess retrieval and answer generation quality. We instantiated this taxonomy for competency questions tailored to SWA research (see Section 10.4.3) based on the concept of our general classification and data schema and the corresponding labeled SWA dataset resulting from our literature study (Chapter 7).

Question Complexity To structure our evaluation, we developed a question taxonomy considering the question complexity dimension based on scientific literature. This allows us to categorize questions (i.e., user input query) and construct a question dataset that contains a fair mixture of different kinds of competency questions (cf. Section 2.8.8) according to their complexity. Therefore, we focus on categories based on question complexity required for answer generation. Drawing from KGQA dataset literature and research question classifications [18, 201, 58, 55, 235, 57, 33, 118], we identified the following eight distinct question categories.

Basic questions enable direct answer retrieval and answer extraction from a KG without requiring additional operations. The answer typically does not necessitate inference, aggregation, or transformation. This category represents the simplest form of retrieval, relying solely on straightforward lookup mechanisms (e.g., "What is the definition of the Client-Server software architecture pattern?"; assuming the answer is stored in a single triple. "What is the replication package link of 'Trace Link Recovery for Software Architecture Documentation'?").

Aggregation questions demand synthesizing multiple information pieces (e.g., "What is the average runtime of systems based on the client-server architecture?" or "Which papers evaluate the property Security and have input data available?").

Counting questions involve enumerating relevant data points (e.g., "How many KGQA approaches were published between 2020 and 2024?" or "How many papers did Manar Mazkatli publish?").

Ranking questions necessitate ordering multiple data points according to specified criteria (e.g., "How do training-free KGQA approaches perform, sorted by F1 score?" or "Which papers evaluate the property Reliability? Rank the papers in descending order by their publication year.").

Comparative questions require evaluating attributes across two or more data points (e.g., "Is method A better than method B?", or "Is machine learning better than information retrieval for trace link recovery?", or "Among publications that use grounded theory for evaluation, what is the number of papers with publicly available input data versus those without?").

Relationship questions require identifying connections or dependencies among multiple entities within a KG. These relationships, such as causalities, associations, or correlations, often span multiple `RDF` triples and are not explicitly stored in a single statement.

Answering such questions involves interpreting the structure and semantics of the graph to uncover implicit relationships (e.g., "What are groups of authors that regularly publish papers about AI in SE together?" or "Among the papers published by the author Mohamed Soliman, what is the distribution of paper classes per publication year?" or "For each year, what is the number of evaluation approaches that have been applied in papers published by Danny Weyns?").

Negation questions require detecting conditions that do not hold, based on explicit negation or absent information (e.g., "Which KGQA approach does not use training?" or "Which papers investigating architecture evolution have not used input data?").

Superlative questions require identifying extremes (qualitative or quantitative terms) among multiple data points (e.g., "Which architecture pattern ensures lowest latency?" or "Which is the evaluation method that has been applied the most to evaluate architecture evolution in papers published in 2020?").

Use Cases In addition to question complexity, we developed use cases based on real-world scenarios to support KGQA dataset construction.

Our knowledge base distinguishes between two metadata types:

(1.1) **conventional descriptive metadata** (e.g., bibliographic metadata such as title, authors, research level, and replication package URL), and

(1.2) **knowledge-based descriptive metadata** (e.g., research object, evaluation method, property; related to an artifact's content data).

The use case design was guided by two categories:

(2.1) (Expected) **Answer Type** specifies whether the expected answer is conventional and/or knowledge-based metadata, and

(2.2) (Input) **Condition Type** indicates whether the question includes constraints based on metadata.

Using these categories, we formulated six use cases tailored to scholarly literature search tasks in SWA research based on the data schema, as depicted in Figure 7.3.

Formulation of Competency Questions To enable meaningful evaluation of KGQA retrieval approaches, the datasets were designed to exhibit high variability across the aforementioned dimensions. Each question is categorized according to the **Question Complexity**, ensuring an even distribution of retrieval complexity. Additionally, the questions are mapped to one of six **Use Cases** to ensure balanced coverage of application scenarios. The dataset also distinguishes between **Semi-typed** and **Untyped Questions** to assess the retriever's ability to handle incomplete or ambiguous type information. Semi-typed questions include partial type annotations within `RDF` triples, while untyped questions omit type references entirely, requiring the system to infer entity types based solely on context. This structure supports comprehensive evaluation of both retrieval performance and semantic robustness (see Section 13.4).

Example for Semi-typed Question: "Which papers applied the *evaluation method* Argumentation in the *publication year* 2020?"

Example for Untyped Question: "Which papers applied Argumentation in 2020?"

10.4. Experiment Preliminaries

In the following sections, we provide an overview of the experiment preliminaries for the contributions in Part III. We begin by presenting the experimental setup and environment (Section 10.4.1). Next, we introduce an analysis of the dataset (Section 10.4.2), which was derived from our literature study (Chapter 7) and is used in the experiments described in Section 12.4. Lastly, we detail the gold standard question–answering dataset for retrieval (Section 10.4.3), which is based on our previously introduced taxonomy (Section 10.3) and used in the experiments presented in Section 13.4.

10.4.1. Experimental Environment

In this section, we present evaluation preliminaries w.r.t. the experimental environment while following the instructions of the best practices in our experiments (cf. Chapter 12 and Chapter 13), as presented in the previous Section 10.2.

Hardware and Software Environment All experiments were executed under a consistent hardware and software environment. The execution server operates on Linux kernel version 6.1.0-23-amd64 and is equipped with an Intel® Xeon® Gold 6258R CPU @ 2.70GHz, supporting 112 parallel threads, a Tesla V100S-PCIE-32GB GPU, and 256 GiB RAM.

All software projects are delivered with a complete list of (package) dependencies and versions. Overall, the software frameworks (i.e., the classification framework in Chapter 12 and the retrieval framework in Chapter 13) were constructed using the LangChain library[9], an open-source framework designed to help developers build applications using LLMs. Additionally, the experimental setup utilizes both local and remote LLMs. Local LLMs are managed using Ollama[10] and HuggingFace Transformers[11], while remote LLMs are accessed via the OpenAI API[12]. We use Optuna as a hyperparameter optimization tool [3].

Experiment Tracking and Evaluation We use Weights & Biases[13] (W&B) as an experiment tracking tool (cf. Section 10.2.1) to systematically log all evaluation results presented in Chapter 12 and Chapter 13. Additionally, we monitor energy consumption and estimate carbon emissions via CodeCarbon (cf. Section 10.2.2). There is an integration[14] of CodeCarbon for W&B that allows users to track the carbon emissions of experiments and logs them directly to the W&B dashboard, alongside other experiment-specific metrics.

ORKG Infrastructure and Application Programming Interface (API) The ORKG is currently deployed and hosted on three distinct environments, each serving a specific purpose within

[9] https://www.langchain.com/, [Last accessed on 2025-09-18]
[10] https://github.com/ollama/ollama [Last accessed on 2025-09-18]
[11] https://huggingface.co/docs/transformers/index [Last accessed on 2025-09-18]
[12] https://platform.openai.com/docs/overview [Last accessed on 2025-09-18]
[13] https://wandb.ai/site/ [Last accessed on 2025-09-18]
[14] https://wandb.ai/amanarora/codecarbon/reports/Tracking-CO2-Emissions-of-Your-Deep-Learning-Models-with-CodeCarbon-and-Weights-Biases--VmlldzoxMzM1NDg3 [Last accessed on 2025-09-19]

the development and operational lifecycle. The *production*[15] environment provides access to the stable, publicly available version of the knowledge graph and reflects the most up-to-date and reliable deployment. The *sandbox*[16] environment serves as an experimental platform, enabling researchers to prototype and test features and tools without affecting production data. Lastly, the *incubating*[17] environment is designated for the early testing of features that are still under active development. All experimental procedures for the retrieval approach in Chapter 13 were executed within the sandbox environment.

To facilitate programmatic access to the different environments, the ORKG backend is exposed through a publicly available RESTful API[18] and an accompanying ORKG's Python package[19]. The API supports the full suite of publicly accessible read and write operations, whereas the Python package serves as a high-level wrapper designed to simplify access for developers and researchers through code. For our evaluation, we used a hybrid strategy.

10.4.2. Gold Standard Dataset for Classification

For our experiments, we need different datasets corresponding to the abstraction levels M0 and M1 (see Section 5.2.2).

In the following, we present an analysis of the gold standard dataset derived from our literature review [153] (see Chapter 7). This analysis specifically focuses on the classification concepts and labels used for **knowledge-based descriptive metadata** that characterize the findings of a paper. These include the categories *Research Object*, *Evaluation Method*, and *Property*. Additionally, we examine **bibliographic descriptive metadata**, considering authorship information. This dataset is used to show the feasibility and maturity of our classification framework (Chapter 12, VQ I-3.2). It comprises publications labeled across multiple classes, reflecting the broad spectrum of research practices (see Chapter 7). Due to the heterogeneity and complexity of knowledge-based statements, a **multi-label classification approach** is required. To support accurate classification, transformer-based models must be adapted with specialized layers capable of handling multiple labels per document.

Knowledge-based Descriptive Metadata To describe a knowledge-based statement in papers based on their *Findings*, we have to consider the categories of *Research Object*, *Evaluation Method*, and *Property*. A paper is assigned one, two, or three ground-truth labels, each belonging to one of these categories. For instance, a paper may have up to 2 labels for *Research Object*, up to 3 labels for *Evaluation Method*, and up to 3 labels for *Property* (first-level aggregation). As mentioned before, we face a multi-label problem, where one paper or a knowledge statement can be assigned to several labels. A major challenge for an automated classification approach arises from the imbalanced distribution of the dataset, as presented in Section 12.2. In the following, we analyze the dataset in more detail.

[15] `https://orkg.org` [Last accessed on 2025-09-18]
[16] `https://sandbox.orkg.org/` [Last accessed on 2025-09-18]
[17] `https://incubating.orkg.org/` [Last accessed on 2025-09-18]
[18] `https://tibhannover.gitlab.io/orkg/orkg-backend/api-doc/` [Last accessed on 2025-09-18]
[19] `https://orkg.readthedocs.io/en/latest/` [Last accessed on 2025-09-18]

The most occurring labels for the **category *Research Object*** are *Architecture Analysis Method* (31 instances), while *Architectural Assumptions* and *Technical Debt* are the least represented (3 instances each), resulting in an **imbalance ratio of approx. 1:10**.

The most occurring labels for the **category *Evaluation Method*** are *Case Study* and *Technical Experiment* with around 50 instances, whereas *Verification* appears only once, yielding an **imbalance ratio of approx. 1:50**.

For the **category *Property***, Product Quality is the most frequent label (107 instances), followed by Quality in Use (59 instances), while Quality of Analytical Method is the least frequent (31 instances). Each data point in the dataset includes all three classes and may contain multiple labels per class, resulting in a multi-class, multi-label classification problem. Product Quality remains the most frequent single label, while combinations such as Quality in Use and Quality of Analytical Method are rare (only two instances). This pronounced imbalance necessitates specialized handling during model training. Standard binary or multi-class classifiers may yield distorted metrics and inaccurate predictions.

Bibliographic Metadata Authorship In addition to the knowledge-based descriptive metadata of research papers, we consider bibliographic metadata, with a particular focus on authorship. This analysis informed the implementation of a customized feature within our classification framework (see Figure 12.2), specifically designed to guide data splitting strategies for fine-tuning language models in classification tasks. Our dataset revealed varying patterns of author occurrence, i.e., some authors appeared only once, while others were associated with multiple papers. This repetition poses a risk that LLMs may learn stylistic (writing) patterns specific to leading authors, rather than generalizable, knowledge-specific content. To mitigate this risk, we applied grouped cross-validation, ensuring that papers by the same author appear exclusively in one fold, either training or testing, to prevent data leakage. To implement this, we constructed an author graph where each paper is represented as a vertex, and edges connect papers sharing at least one author. Disconnected subgraphs were then used to define **author-based splits** in transfer-learning approaches, enabling robust evaluation while minimizing stylistic (writing) bias.

10.4.3. Gold Standard Question-Answering Dataset for Retrieval

After following a systematic approach for the creation and formulation of competency questions according to our taxonomy, as presented in Section 10.3, we constructed a corresponding QA dataset for the parameter selection and the evaluation of retrieval and answer generation quality of our retrieval approach and the baselines. The QA datasets are provided as `.csv` files; one for each ORKG knowledge base. One knowledge base captures one of the four graph variants, GV1, GV2, GV3, and GV4 (see Section 11.3), based on the same classification and data schema for SWA research. The header entries in the respective files are organized as follows:

- **UID**: A unique identifier that distinctly identifies a specific question–answer pair.
- **Competency Question**: A question formulated in natural language and submitted to the QA system.

- **Topic Entity**: An optional entry point within the corresponding target RKG, used by certain retrieval algorithms. This entry is optional, as not all retrieval approaches require such information.
- **Golden Answer**: A reference answer, formulated in natural language, that is considered correct for evaluation purposes.
- **Golden Triples**: A set of triples contained in the target KG that represent the factual basis for the Golden Answer.
- **Source IDs**: Identifiers of the source publications (literature reference) from which the Golden Answer and Golden Triples were derived.

We used **44 questions for the parameter selection** (Section 13.4.1.2) and **170 questions for the evaluation** of our retrieval approach and the baselines (Section 13.4.2).

10.5. Conclusion

In this chapter, we outlined the idea of our approach, called KARAGEN, comprising a knowledge description and knowledge access module. KARAGEN integrates the complementary strengths of LLMs and KGs within a graph-based RAG-inspired pipeline.

To ensure the development of sustainable AI-based software solutions, we provide best practices and techniques for the experimental design and settings. Furthermore, we introduce evaluation preliminaries by analyzing and presenting our datasets used for the evaluation of an automated classification (Chapter 12) and a retrieval approach (Chapter 13). In particular, we propose a question-answering retrieval taxonomy to derive various competency questions (see Section 2.8.8) that enable a nuanced analysis of system performance w.r.t. the question and retrieval complexity and the diversity of descriptive metadata types. This evaluation strategy (see extrinsic evaluation strategy in Section 5.3) also supports a range of use cases and provides insights into the robustness and adaptability of the proposed retrieval approach.

11. Formalization of an ORKG Template for Software Architecture Research

This chapter builds on the publications "Structuring Scientific Knowledge in Software Engineering Using the Open Research Knowledge Graph: A Use Case and Experience Report" [130] published at the Working Workshop on Research Data Management for and in Software Engineering (RDMxSE'25) and "HubLink: A Novel Question Answering Retrieval Approach over Knowledge Graphs" [134] published at the 24th International Semantic Web Conference (ISWC2025), Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG). We ask the following research question along with its corresponding validation question:

RQ II-1: How to map our developed general classification and data schema for software engineering research to the concepts of ORKG templates, considering state-of-the-practice and normative aspects?

VQ II-1: What are the different ORKG template variants of our developed classification schema for software architecture research to derive a populated knowledge base?

In this chapter, we aim to explore the capabilities of ORKG for semantic structuring in Software Engineering (SE) research, with a particular focus on Software Architecture (SWA) research. To this end, we use our general classification and data schema based on the classification concepts *Findings*, *Validity*, and *Evidence*, instantiated for SWA research objects (cf. Chapter 7) to derive an ORKG template for practical use (Section 11.2). Building upon this template, we further derive three template variants based on two orthogonal characteristics, namely path length and contribution granularity, to construct, in total, four knowledge bases (Section 11.3) for our experiments presented in Chapter 13. This setup enables us to evaluate the robustness of our retrieval approach (see Section 13.2), considering both structural and lexical variability across different graph variants.

Additionally, we present our serverless web application *Visulite* (Visualization of Literature), which enhances our replication package data [155] with different visualization techniques and table filter options to browse through the gold standard dataset and export user-defined research overviews for comparisons (Section 11.4).

Consequently, this chapter constitutes the contribution C5.

11.1. Problem Statement and Research Aim

The effective structuring and dissemination of research knowledge in SE remains a challenge, where empirical findings are often accompanied by complex research and replication

artifacts (see Section 1.1). Traditionally, scientific papers serve as the primary medium for communicating research outcomes. However, this format limits machine-readability, machine-actionability, and reusability; key requirements for modern research infrastructures aligned with the FAIR principles (Section 2.1.3).

To address these limitations, we adopt machine-readable formats, as part of contribution C2 (Chapter 6), enabling documentation, traceability, and evolution support (cf. Chapter 8). However, to ensure broader visibility, community engagement, and sustainable reuse, a robust and accessible knowledge base hosted within a FAIR-compliant infrastructure is required.

In this context, we explore the use of the ORKG as a semantic infrastructure for structuring research artifacts. Specifically, we investigate the use of ORKG templates as reusable blueprints for encoding domain-specific knowledge, such as in SWA research. These templates have to serve both descriptive (state-of-practice) and normative functions. The normative dimension, in particular, holds the potential to improve and systematically shape research practices in the context of meta-research. Specifically, the templates should aim to reflect current research workflows while simultaneously guiding the community toward standardized semantic representations and normative reporting conventions. Moreover, we aim to set an initial standard for the SWA research community, supporting digital transformation in research (see Section 1.2).

To support multi-modal knowledge graphs, we introduce *Visulite*, a lightweight tool that enables the visualization of replication data and the integration of structured data visualizations and comparisons (e.g., charts, tables) into the ORKG. This enhances the expressiveness of research representations and supports diverse data modalities.

Furthermore, we define multiple graph variants based on our classification and data schema for SWA research. These variants enable the evaluation of schema-agnostic retrieval approaches, allowing us to assess how different semantic structures affect retrieval performance. Our open-access repository of KARAGEN documents the evolution of these templates, supporting traceability, benchmarking, and collaborative refinement.

Through this work, we aim to foster community-driven development of semantic research infrastructures in SE, especially in SWA research, promote the reuse of structured knowledge artifacts and knowledge statements, and support the empirical evaluation of retrieval strategies across heterogeneous graph representations.

11.2. Semantic Structuring for Software Architecture Research

In this section, we argue about our consideration to derive an ORKG template for SWA research in practical use by answering our research question RQ II-1: *How to map our developed general classification and data schema for software engineering research to the concepts of ORKG templates, considering state-of-the-practice and normative aspects?*

The classification and data schema for SWA research (see Section 7.4) was systematically mapped to the ORKG template concept through a linear transformation process. This mapping involved the identification of relevant entities and relations, leveraging the

hierarchical structure of the schema and a finite set of standardized instances for each core classification concept (e.g., paper class). To ensure semantic richness and avoid information loss, relationships and cardinalities were carefully modeled, resulting in a structured yet complex representation of research findings as knowledge statements. The concrete data extraction format used in the literature review (see Table 7.1) provides a detailed overview of the schema and its implementation.

The overarching objective was to develop a domain-specific ORKG template for SWA research that offers long-term value across three dimensions:

1. Maintainability: supporting schema evolution and versioning,
2. Ease of Use: ensuring readability and clarity for ORKG contributors, and
3. Suitability for Purpose: enabling effective and efficient knowledge retrieval.

The mapping process was guided by the following structural principles:

- Organization of contribution descriptions based on core knowledge statements (findings, validity, evidence),
- Differentiation between descriptive and normative representations, capturing both observed practices and idealized recommendations,
- Use of finite option sets (enumerations), derived from standards, community consensus, or expert knowledge (e.g., research object categories), and
- Modeling of complex data structures, including cyclic relationships and multidimensional cardinalities (e.g., research object, evaluation, and property statements).

A key modeling challenge was the **representation of knowledge statements** that encompass research findings, associated validity evidence, and threats to validity. In SWA literature, we observed that, e.g., threats to validity, tool support, and input data are typically discussed at the level of the entire publication, rather than being attributed or linked to individual findings or research objects. This raises the question of whether findings and their supporting validity should be encapsulated within a single contribution type or distributed across multiple types to allow flexible representation. Consequently, a hybrid model was proposed: contributors can choose to insert knowledge statements (findings, validity, evidence) either within a unified contribution type or across distributed types. This flexibility enables the representation of multiple findings and their associated validity and evidence at the paper level (reflecting the state-of-practice) or at a more granular level (reflecting normative modeling). This approach supports a rigorous yet adaptable representation of research practices in SWA.

Another major challenge was the representation of **multi-class and enumerated categorization types**, such as paper classifications (e.g., evaluation research, solution proposals, and validation studies). As ORKG templates currently lack native support for multiple-choice data types, alternative modeling strategies were required. Two strategies were considered. The first involved using plain text input, which offers flexibility for users to enter arbitrary values. However, this method lacks constraints on input consistency, potentially leading to discrepancies in data representation, such as variations in casing, typographical errors, or synonymous terms, thereby undermining semantic uniformity across contributions. The second approach employed a list of Boolean values, wherein each predefined category is represented as a Boolean property indicating its selection status. This method enables template authors to enforce a controlled vocabulary, promoting consistency. Nevertheless, it poses limitations in terms of extensibility: any future additions

to the list would result in missing values for previously annotated contributions, as these new options would not have been available at the time of initial data entry. To balance consistency and flexibility, both approaches were applied selectively based on the nature of the metadata being modeled. The Boolean list method was adopted for classification concepts that are relatively stable and well-established, such as paper class, threats to validity, and availability statements for evidence. Conversely, string-based representations were used for concepts expected to evolve more dynamically.

Furthermore, to capture **complex interrelations** among research objects, evaluation methods, and properties, a nested structure was introduced within the template. This design facilitates the representation of multidimensional research findings without semantic loss, thereby supporting both human interpretability and machine-based reasoning.

Figure 2.2 presents an overview of our derived ORKG template for SWA research.

11.3. Knowledge Graph Variants and Population

To semantically represent scholarly contributions in the ORKG, researchers and users can add publications by providing a DOI or title. The system either links to an existing resource or creates a new one, automatically populating bibliographic metadata of research artifacts. To capture the substantive content of a publication (i.e., knowledge-based descriptive metadata), users employ the *Contribution* content type (see Section 2.6), which allows the structured addition of research findings. For our experiments requiring a knowledge base (Section 13.4), we integrated labeled data into the ORKG by designing custom templates for individual contributions. Following, we address our validation question VQ II-1.

To evaluate the robustness of our retriever under varying graph structures, we constructed four RKG variants, reflecting our classification and data schema for SWA research, based on two orthogonal characteristics: *path length* (i.e., affecting the number of hops) and *contribution granularity* (i.e., affecting the degree of branching).

The **Path Length** determines the semantic depth of graph traversals. Long paths preserve rich semantic relationships through detailed intermediate nodes, enabling more expressive and context-aware queries. However, they require more complex traversal operations. In contrast, shorter paths simplify retrieval by collapsing relationships into direct links, potentially sacrificing semantic richness.

The **Contribution Granularity** affects how research contributions are represented. Distinct contribution nodes create separate graph entities for each contribution within a paper, maintaining clear boundaries between different research outputs and their associated metadata. Cumulative contribution nodes in the graph consolidate all knowledge-based metadata from a paper into a single node, reducing graph complexity but potentially blending distinct research concerns.

These aforementioned dimensions reflect trade-offs between semantic expressiveness and structural clarity, and are crucial for evaluating schema-agnostic approaches for knowledge graph querying and reasoning. Combining these characteristics yields four evaluation graph variants:

Graph Variant 1 (GV1) contains long paths with distinct and distributed contribution nodes (maximal semantic richness), see Figure 11.1

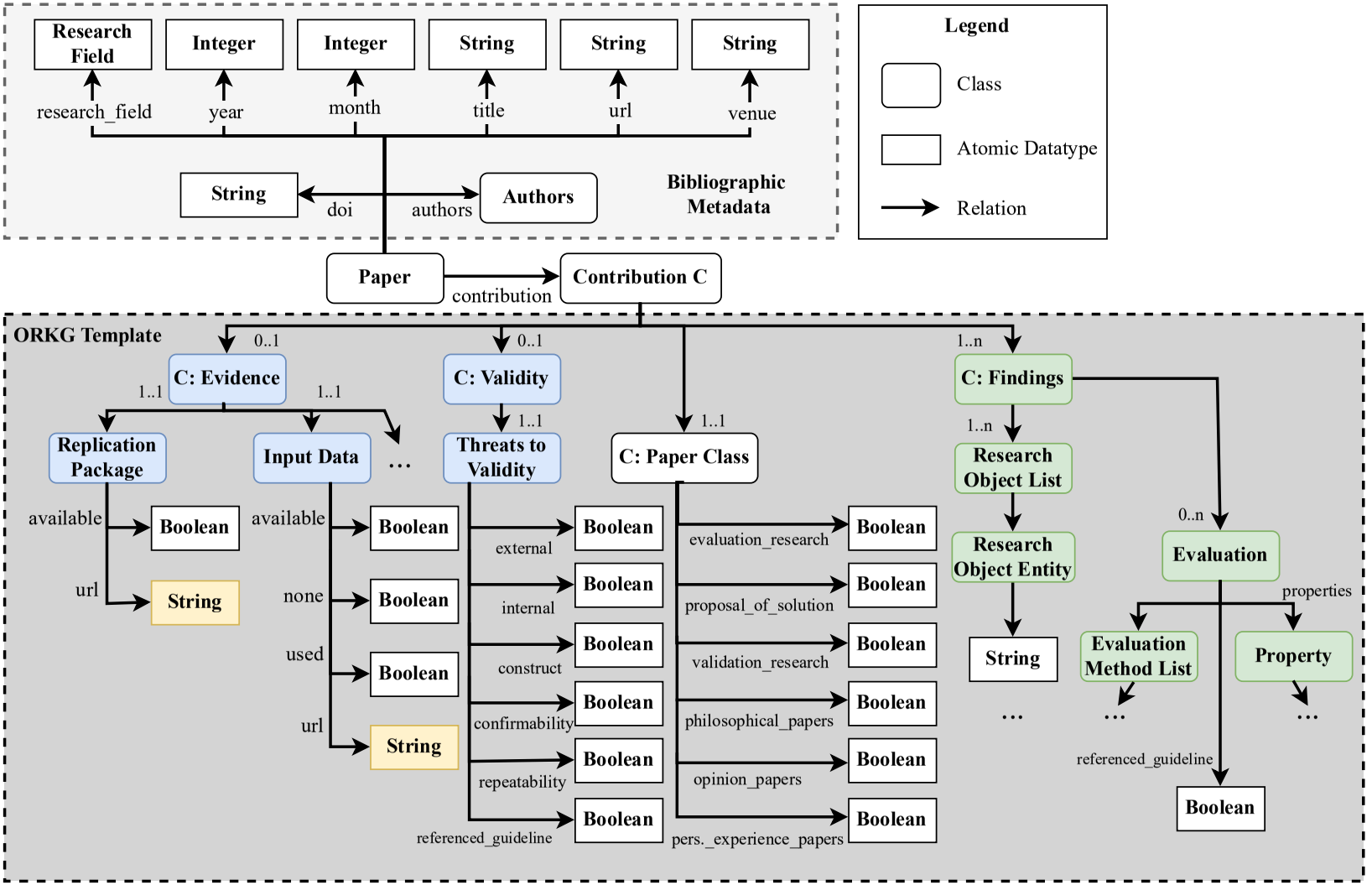

Figure 11.1.: An ORKG template based on the classification and data schema for software architecture research [153] (Figure adapted from Kaplan et al. [130])

Graph Variant 2 (GV2) has long paths with cumulative contribution nodes (centralized in a single contribution and balanced semantic depth)

Graph Variant 3 (GV3) has short paths with distinct contribution nodes (simplified structure and preserved granularity)

Graph Variant 4 (GV4) contains short paths with cumulative contribution nodes (minimal complexity)

These variants allow us to assess how structural differences affect retrieval performance and whether the retrieval approach remains schema-agnostic, i.e., applicable across graph variants based on the same classification and data schema.

In the appendix, we provide an overview of the four graph variant templates using an excerpt of our classification and data schema (*Research Object* and *Input Data*) in Figure A.1.

11.4. Visulite: Visualization of Literature

In addition to supporting textual data types and semantic classes, the ORKG enables the integration of visualizations within its templates and ORKG comparisons. This multimodal capability enhances the expressiveness of research knowledge representations, particularly for illustrating empirical results. One of ORKG's core features, i.e., ORKG comparisons (see Section 2.6), offers tabular overviews of state-of-the-art research on specific questions and research problems, which can be enriched with visual elements to improve interpretability.

As part of our work on a structured classification and data schema for SWA research [153], we developed *Visulite*[1], a lightweight, serverless web application designed to visualize replication package data (e.g., our replication package for the SWA dataset [155], derived from the literature review study in Section 7.6). Figure 11.2 depicts a screenshot of the corresponding web interface. Visulite supports various chart types (e.g., bar charts, bubble charts) and interactive tables with filtering options, facilitating intuitive exploration and representation of raw research data. These visualizations can be embedded into ORKG services to complement textual and tabular content.

Visulite is designed with flexibility and user requests in mind. Users (researchers) can customize visual outputs by selecting relevant data fields and adjusting table columns to reduce complexity and focus on essential information. The application supports data export in multiple formats, including `.svg` for charts and `.csv` for tabular data, enabling persistent and reusable visual artifacts.

From an SE perspective, Visulite emphasizes and supports modularity and extensibility. Its internal data model is decoupled from both input formats and resulting output representations, allowing for the integration of additional parsers beyond the default *BibTeX* parser. This design supports diverse input sources and facilitates the reuse and extension of existing visualization components. The use of the D3.js[2] library further enhances customization, enabling fine-grained control over chart aesthetics and user interface elements.

[1] `https://softwarearchitectureresearch.github.io/visulite/` [Last accessed on 2025-10-12]
[2] `https://d3js.org/` [Last accessed on 2025-10-11]

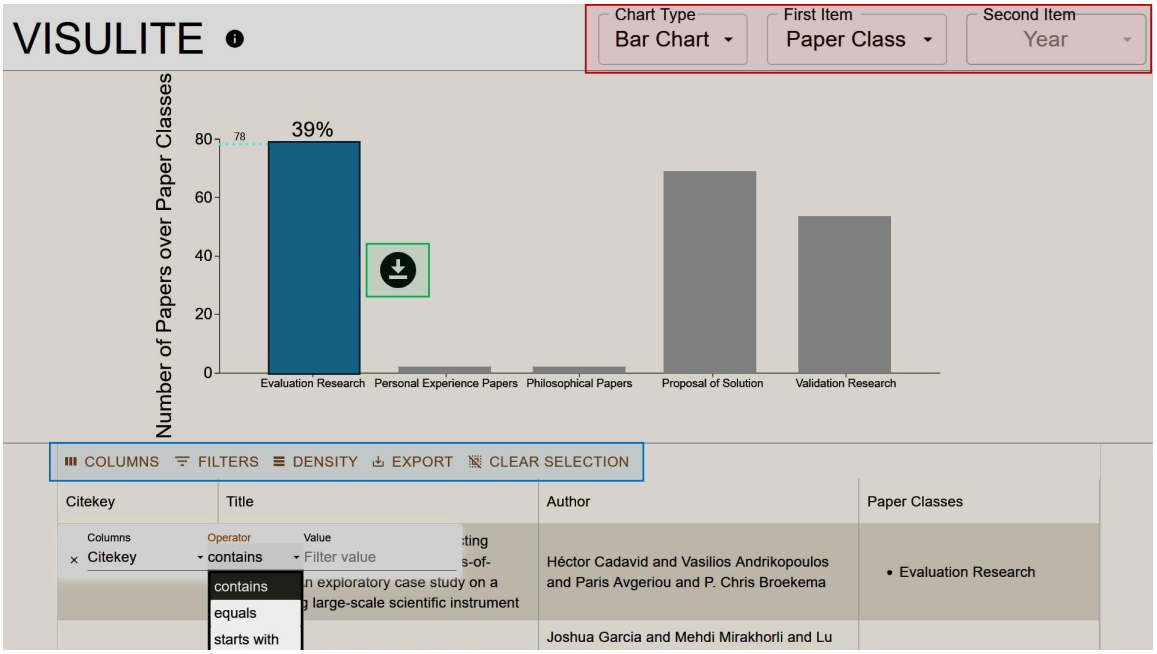

Figure 11.2.: Web interface of Visulite [130]. Colored boxes denote core tool features: settings for charts (red box), *.svg* download option (green box), and a table with filtering and export options (blue box).

We invite the research community to adopt and extend Visulite for their own use cases. The source code and documentation are available in the Visulite GitHub repository, including setup instructions and usage examples in the `README` [42].

11.5. Conclusion

This chapter addressed key limitations in the (availability and) use of machine-readable formats for SE research knowledge within a FAIR-compliant infrastructure. In response, we explored the capabilities of the ORKG to derive a reusable template that serves as a blueprint for our general classification and data schema (see Chapter 7). This template enables structured comparisons across research artifacts and supports the expression of both current state-of-practice and normative aspects. This demonstrates the value of analyzing structuring approaches for SE research artifacts to reflect and improve current research practices. Using the derived and resulting template, we constructed a knowledge base for SWA research, capturing validated knowledge statements from scholarly publications. Additionally, we also derived three further ORKG template variants aligned with our classification schema to evaluate how different representations of the schema affect retrieval performance and influence knowledge access (see Chapter 13). To enhance the text-based representation in the RKG, we developed our serverless tool *Visulite* to provide (i) visualizations of the replication package and the knowledge base, supporting various chart types, and (ii) a table with filtering options to browse the extracted knowledge-based metadata of SWA research artifacts (paper, data, software). Together, these contributions

support the advancement of FAIR principles in SE research and offer practical tools for improving the accessibility, comparability, and reuse of structured knowledge.

12. A Classification Framework for Research Artifacts

This chapter builds on the publications "A Classification Framework for Scientific Documents to Support Knowledge Graph Population" [131] published at the Working Workshop on Research Data Management for and in Software Engineering (RDMxSE'25) and "Responsible and Sustainable AI: Considering Energy Consumption in Automated Text Classification Evaluation Tasks" [132] published at the IEEE/ACM International Workshop on Green and Sustainable Software (GREENS@ICSE2025). We ask the following research question along with its corresponding validation questions:

RQ II-2: How to construct a semi-automated classification framework using natural language processing techniques (differently combined) for classifying scientific papers (in software architecture research)?

VQ II-2.1: How well can automated approaches classify software architecture papers w.r.t. the data schema in Chapter 7 according to the labels *Research Object*, *Properties* (second-level properties), and *Evaluation Method*?

VQ II-2.2: What are suitable configurations for the classification task, especially regarding performance and energy efficiency?

This chapter presents a modular framework designed to support researchers and developers in deriving and selecting an appropriate automated classifier for multi-label classification tasks. In particular, the framework facilitates the categorization of research artifacts, primarily scientific papers, based on a predefined classification and data schema. Building upon discussions of the problem statement (Section 12.1) and the key challenges (Section 12.2) associated with an automated multi-label classification approach, we derive essential design considerations for the framework. This includes architectural principles and design patterns, as well as capabilities and features to guide the framework's development (Section 12.3).

To assess its feasibility, we evaluate the framework by applying two classification strategies (i.e., fine-tuning and prompting) within a classification use case focused on Software Architecture (SWA) research. For this, we use the labeled dataset from our literature study (cf. Chapter 7) as the gold standard for the experiments presented in Section 12.4. Our framework considers both performance and sustainability aspects (in terms of energy consumption) to enable benchmarking and trade-off decisions for researchers and developers. As a consequence, they can derive a suitable classifier that can be embedded into real-world applications. The evaluation follows the automated strategy of our evaluation approach, as introduced in Section 5.3.

We publicly provide our framework through an open-access repository, thereby supporting the principles of Open Science and the FAIR guidelines (see Section 2.1.3). This promotes collaboration, community engagement, and reusability, while contributing to the continued development and maturation of the framework.

The purpose of using our developer and evaluation framework is twofold:

- Derive a suitable classifier to enable semi-automated knowledge graph population in the publication phase (cf. bridge between *Documenting Research* and *Organizing Research* in Figure 1.1) for individual research artifacts (papers) based on a predefined classification and data schema provided by stakeholders (e.g., researchers as authors).

- Derive a suitable classifier for stakeholders (e.g., researchers and practitioners) to update systematic literature studies, especially systematic mapping studies, in the designing and conducting phase (cf. *Performing Research* in Figure 1.1) in a semi-automated way based on a previous classification schema that can evolve.

Consequently, this chapter constitutes the contribution C6.

12.1. Problem Statement

The rapid growth of SE research publications results in an amount of scholarly knowledge that needs to be effectively and efficiently managed, organized, and communicated in digital libraries and scientific search engines (cf. Section 1.1). For this, we previously discussed the key benefits of knowledge graphs (KGs) to accomplish this task (cf. Section 10.1).

Research Knowledge Graphs (RKGs), capturing metadata from research artifacts (papers and replication artifacts comprising software and data), can be populated either manually or through automated methods, each offering distinct advantages and limitations. Manual population, such as the initially intended process in ORKG (cf. add paper function[1]), is typically performed by domain experts and ensures high precision, semantic accuracy, and alignment with established ontologies and data schemas based on expert knowledge or established and evolving standards [14]. It allows for nuanced interpretation of complex scientific content and supports rigorous quality control. However, this approach is time-consuming, resource-intensive, and difficult to scale, making it less feasible for large or dynamic datasets, as in academic settings. In contrast, automated population, often based on natural language processing and machine learning techniques, enables efficient, large-scale classification and extraction of information from research artifacts. It supports timely updates and reduces the cost of curation, but may suffer from lower accuracy, especially in handling domain-specific language and semantic ambiguity. Automated methods also face challenges in aligning classified and extracted information with formal ontologies and data schemas without human oversight. As a result, hybrid approaches that combine automated extraction with manual validation by domain experts are increasingly adopted to leverage the strengths of both strategies, achieving scalability while maintaining a high standard of quality [14].

[1] `https://orkg.org/papers/create` [Last accessed on 2025-09-19]

The timing of metadata collection from research artifacts plays a critical role in the organization, discovery, and reuse of scholarly knowledge. This typically includes descriptive metadata, especially referring to a research artifact's content, such as the research object of a scientific paper. The quality and completeness of metadata directly influence the visibility, interoperability, and long-term preservation of scientific knowledge and outcomes. As a result, a systematic collection of metadata related to papers and replication artifacts may support evidence-based studies in SE in a decentralized and sustainable manner, while reducing manual effort currently required from small author groups. Metadata can be collected at different stages of the research lifecycle (cf. Figure 1.1), most notably during the pre-publication and post-publication phases (bridging publishing and organizing research), each with its own advantages and limitations.

To support hybrid knowledge graph population and metadata collection in the publication phase to contribute to evidence-based studies based on written research artifacts, we implemented a developer framework to assist in deriving suitable classifiers for these tasks and integrating them into real-world applications and (future) infrastructures, such as the ORKG. We elaborate the key features of this framework from a discussion about challenges associated with automated approaches in academic contexts, which are discussed in detail in the following section.

12.2. Challenges for an Automated Approach

In this section, we discuss potential challenges associated with automating multi-label classification tasks within our classification framework. The discussion is based on general and preliminary considerations aimed at selecting an appropriate architectural design that aligns with our research objectives; specifically, the development of a classification framework that supports both reusability and extensibility.

Degree of Complexity Describing research artifacts w.r.t. their different kinds of descriptive metadata presents a challenge due to the inherent heterogeneity and semantic complexity of the involved metadata. This complexity is amplified when the task is formulated as a multi-label classification problem, wherein each artifact may simultaneously belong to multiple, non-mutually exclusive categories (classification dimensions). Standard libraries and toolkits often provide limited support for such multi-label scenarios, as they are typically optimized for single-label or multi-class classification tasks. As a result, specialized methods and preprocessing steps are required to effectively model and evaluate multi-label classification in the context of descriptive research metadata.

Missing Representatives in the Dataset The absence of representative samples in a dataset presents a significant challenge when developing and evaluating automated classification approaches. Consequently, the proposed framework has to incorporate classification strategies capable of addressing such deficiencies by compensating underrepresented classes. To facilitate effective fine-tuning with limited class representatives, the framework should be able to classify data instances at a higher level of abstraction within the classification

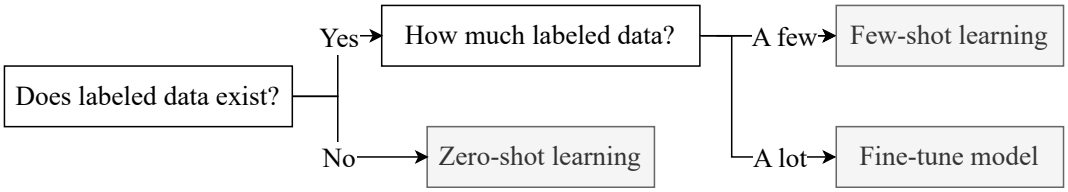

Figure 12.1.: Overview of suitable classification approaches based on the amounts of labeled data (Figure adapted from Tunstall et al. [291, p. 250])

hierarchy. This approach increases the number of available representatives per category, thereby improving the robustness and generalizability of the classification approach.

(Highly) Imbalanced Dataset A dataset can be (highly) imbalanced, i.e., the number of instances in one or more classes is considerably higher than in the others (cf. Section 10.4.2). This is most relevant in classification tasks, where the model learns to distinguish between multiple categories or labels/classes. In this case, the distribution of multiple class labels is biased, indicating the certain classes have far more instances (majority classes) than others (minority classes). To address this problem, the framework should support different data augmentation techniques such as resampling (i.e., oversample minority or undersample majority classes) and perturbation strategies. In addition, the framework should enable various data splitting strategies to ensure robust model evaluation. For instance, our framework should include a bibliographic metadata-driven splitting mechanism that partitions the dataset based on paper authorship. This includes that an author appears either in the training, validation, or test split to promote the robustness and generalizability of the resulting classifiers.

Evolution Scenarios w.r.t. Data Schema As research fields evolve, the representation of knowledge and the underlying classification and data schemas associated with research artifacts are also subject to change. Consequently, the framework has to be designed to support changed and extended labeled datasets, enabling adaptability to diverse and evolving (hierarchical) schema definitions across different research contexts to derive domain-specific classifiers.

Generalizability Different research fields in SE could require the use of diverse datasets and corresponding data schemas. Therefore, the framework should be designed with flexibility in mind, allowing for seamless integration of heterogeneous and hierarchical data schemas and datasets to support the specific requirements and heterogeneity of various SE subfields (and beyond).

Suitable Classification Approach The framework should be able to support multiple classification approaches (i.e., fine-tuning and prompting) based on the different characteristics of the dataset, as depicted in Figure 12.1. Additionally, it should include result tracking for benchmarking purposes. This implies tracking various metrics for performance, time

(latency), and sustainability aspects (i.e., trade-off between performance and carbon emission, as described in Chapter 5). While metrics for performance are commonly used and reported, sustainability aspects are often overlooked. However, classifiers with comparable performance may differ substantially in their environmental impact. At scale, such differences can become significant [132]. Therefore, sustainability aspects in terms of energy consumption in the inference phase are an integral component of our framework, enabling ecologically informed decision-making when selecting classification models.

Experiment Tracking Logging and experiment tracking are important w.r.t. the different assets, as described in the best practice for experiments in Section 10.2. In this way, we can facilitate transparency and structured comparisons across models, datasets, training conditions, and classification results, enabling evidence-based selection of model configurations for developers. Therefore, our framework should integrate experiment tracking tools and be able to add logging tools.

Open Science Principles We value Open Science principles. Therefore, our framework should be designed in a way to support various types of open-source models and publicly available models with open weights, such as LLaMA [287]. Additionally, it facilitates benchmarking and comparison with proprietary models from providers such as OpenAI, thereby promoting transparency, reproducibility, and a fair evaluation of model performance across different model families (cf. Table 2.2).

12.3. Classification Framework

In this section, we present our framework supporting different automated classification approaches (cf. Section 2.3.1) for scientific documents using LLMs. The framework addresses the key challenges identified in Section 12.2 and serves to answer our research question:

RQ II-2: How to construct a semi-automated classification framework using natural language processing techniques (differently combined) for classifying scientific papers?

Architecture and Framework Features The selection of suitable patterns and an appropriate architecture is crucial to ensure maintainability and reuse of our classification framework. Therefore, the framework adopts a modular architecture. It supports diverse datasets and schemas, including hierarchical ones, and classification approaches; LLM versions/variants; experimental configurations, including data augmentation and sampling strategies; and different dataset splitting strategies for model training (e.g., based on bibliographic metadata, considering authors) to address the aforementioned challenges (see Section 12.2). The framework is based on the *Blackboard* Pattern, comprising a shared data space (blackboard), a controller to manage execution, and agents that process and contribute data to the blackboard. This pattern enables the integration of diverse agents and heterogeneous components, supporting multiple variation points within the experimental setup. Thus, the components of the framework are divided into: *Blackboard* (superordinate concept), *Data*, *Classifier*, and *Utilities*. Each component (module) in the framework is assigned a distinct responsibility and communicates through clearly defined interfaces,

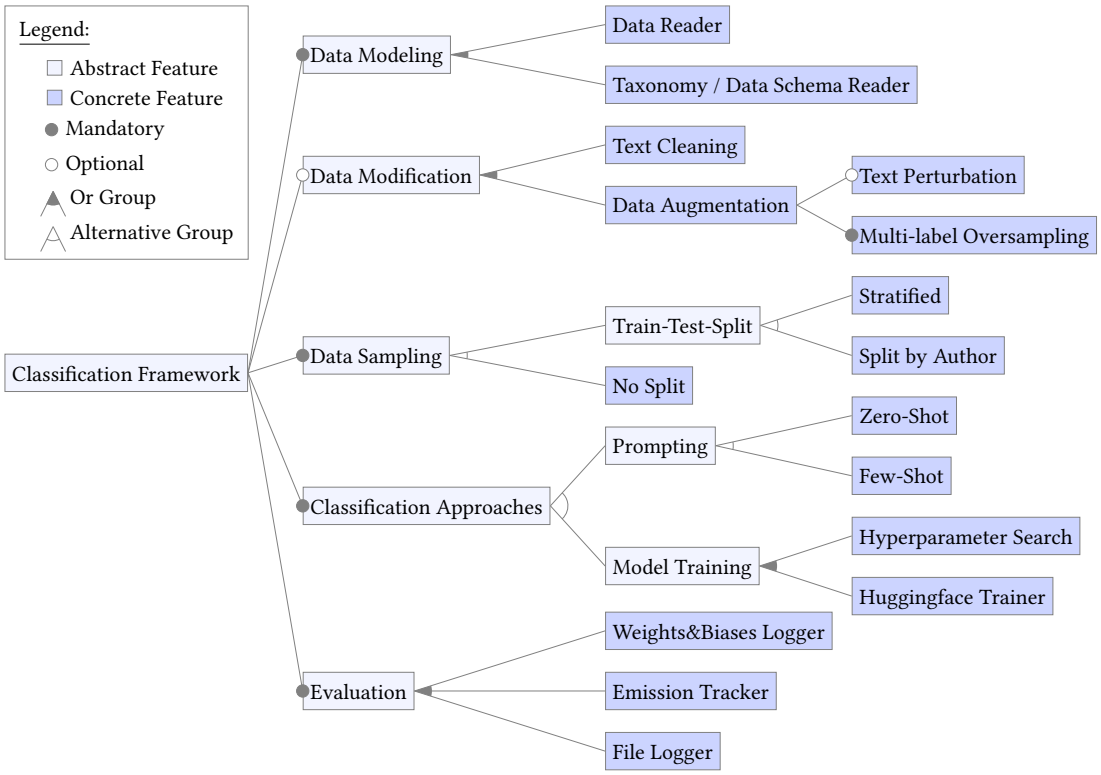

Figure 12.2.: Feature diagram illustrating the current state of the text classification framework, including variation points for different experimental configurations. (Figure adapted from Kaplan et al. [131])
The constraints include: Text Perturbation implies Multi-label Oversampling; Model Training implies Train-Test-Split; Prompting implies No Split.

minimizing dependencies. This modular design ensures that components can be easily replaced, adapted, or added as requirements evolve. As a result, the framework supports the flexible implementation of various features, as illustrated in Figure 12.2.

Following, we present the core features and components of our classification framework that enable to derive and benchmark of various classifiers for hierarchical data schemas.

The **Data** component encapsulates all relevant agents required to prepare the dataset as input for the LLMs. This includes all agents for the data creation, the augmentation (i.e., different data augmentation techniques like text perturbation via TextAttack[2] and multi-label-oversampling strategies), the data handling and splitting (e.g., author split) modules for manipulating the datasets before handing them over to the *Classifier*.

The **Classifier** component encapsulates all agents required for executing classification tasks, including prediction, training, hyperparameter optimization (via Optuna [3], applying Bayesian optimization), and prompting. It supports the specification of search spaces, trial counts for optimization, and the selection of the optimization performance metric (e.g., the macro-averaged F_1 score). For the classification tasks, the framework integrates LLMs from the Hugging Face Transformers, Ollama, and OpenAI libraries (Section 10.4.1).

The **Utilities** component encapsulates, among other things, the data schema and description handling (taxonomy file containing the specified hierarchy and all available labels with their descriptions; see structure in Listing 12.1), and the evaluation logging module via Weights & Biases as experiment tracking tool (see Section 10.2.1).

Definition of Classification Schemas (Data Modeling) One input of our approach is a definition of a classification schema, typically structured as a hierarchical taxonomy. As shown in Listing 12.1, our input format is specified using JavaScript Object Notation (`JSON`). This format enables the ingestion of any classification schema that follows a tree-like structure, where each node may contain an arbitrary number of child elements. Each element in the schema is accompanied by a description that conveys its meaning and purpose. This structured knowledge serves as input to LLMs, allowing them to classify nodes at various levels of the hierarchy based on their contextual understanding.

Listing 12.1: Example excerpt of the taxonomy file to define classification schemas [131]

```
{ "Category 1": {
    "description": "description of Category 1",
    "sub_elements": {
      "Label 1.1": {
        "description": "description of Label 1.1",
        "sub_elements": {
          "Label 1.1.1": {
            "description": "description of Label 1.1.1": { ... }
            }
        }}}},
        ...
}
```

[2] TextAttack 0.3.9 Documentation: `https://textattack.readthedocs.io/en/latest/apidoc/textattack.augmentation.html` [Last accessed on 2023-05-12]

Data Modification The framework includes modules for **text cleaning** and **data augmentation** to address class imbalance and improve model performance. Text cleaning removes special characters and applies stemming, lemmatization, and tokenization. Data augmentation supports multi-label oversampling and text manipulation via TextAttack to enrich the dataset with additional synthetic papers. When TextAttack is enabled, oversampling is mandatory to ensure a balanced label distribution.

For multi-label classification, we provide a **multi-label oversampling** technique that iteratively selects samples to minimize the standard deviation across label distributions. This approach ensures a more balanced representation of minority classes relative to the majority class. These augmentation strategies support the creation of a more representative training set, enhancing the robustness of classification models in the context of software architecture research. To improve dataset quality, our framework combines multi-label oversampling with **token perturbation** techniques to generate synthetic samples. Given the domain-specific nature and editorial quality of software architecture papers in the gold standard dataset, we selected perturbation methods that preserve semantic integrity. Using the TextAttack library, we apply controlled augmentations. For this purpose, we use the WordNetAugmenter, which introduces synonyms based on WordNet, ensuring minimal disruption to meaning. This combined approach enhances data diversity while maintaining the contextual accuracy required for scientific text classification.

Data Sampling The framework supports flexible data sampling strategies, including the option of using no data split (e.g., for prompting classification approaches) or applying data splits. Data splitting can be stratified or based on authorship to support metadata-based evaluation (see Section 10.4.2). To improve generalization and avoid overfitting to author-specific writing styles, grouped cross-validation is applied, ensuring that each author appears in only one fold and preventing overlap between training and test sets.

Automated Classification Approaches To support the diverse and evolving landscape of AI for Research and Science (AI4Research/Science; see Section 1.2), our framework is designed to support two complementary (yet contrasting) LLM-based classification approaches: model training/fine-tuning and prompt-based inference (currently including zero-shot and few-shot). By integrating these two paradigms, the framework ensures flexibility in handling varying levels of labeled training data availability (see Figure 12.1).

Evaluation The framework integrates an **evaluation module**, comprising logging capabilities via Weights & Biases, carbon emission tracking using the CodeCarbon project, and local logging through a file-based logger. These features ensure reproducibility, traceability, and sustainability in the experimental process.

12.4. Evaluation

In this section, we address the validation questions VQ II-2.1 and VQ II-2.2 for the fine-tuning and prompting experiments. Therefore, we evaluate the approaches with the automated evaluation strategy of our evaluation approach (cf. Section 5.3) to assess how the

different automated approaches can accomplish the classification task and to demonstrate the feasibility and maturity of our classification framework. We first describe our evaluation setup (Section 12.4.1), then present the results of both validation questions in (Section 12.4.2) for each classification approach and each classification label (research object, evaluation method, and high-level properties).

12.4.1. Evaluation Settings

In this section, we describe the evaluation settings for evaluating the model training and the zero-shot approach with our classification framework. First, we present the settings for the model training, including the selection of language models, the hyperparameter search space and the results of the hyperparameter search using Optuna (Section 12.4.1.1). Second, we present the settings for the zero-shot prompting, including the selection of language models, hyperparameter settings, and prompt optimization (Section 12.4.1.2). For the evaluation, we used two datasets: (1) the original one from the literature study (Chapter 7) and (2) its extended version (augmented version). We presented the dataset analysis (original dataset) in Section 10.4.2, which indicated a highly imbalanced dataset for all categories (*Research Object*, *Evaluation Method*, and *Property*) and missing representatives.

12.4.1.1. Settings for Model Training

Following, we present the evaluation settings for fine-tuning and the model training.

Selection of Language Models For the hyperparameter search and the fine-tuning of language models we use pre-trained models. We use the following different language models for our experiments: (1) BERT-Base Uncased (Uppercasing not relevant) [54], (2) DistillBERT [248], (3) DeBERTa [98], (4) SciBERT [25], and (5) XLNet [320].

Run Types The following results are categorized by run type: validation performance during hyperparameter search (hp search), test set performance on the trained models (*test*), and evaluation on the author-based data split (*author*).

Hyperparameter Search and Search Space Our classification framework supports hyperparameter optimization using Bayesian optimization via Optuna [3]. To optimize model performance, we conducted a hyperparameter search considering several key parameters. The hyperparameters included: (1) Network initialization seed (*seed*), (2) Learning rate (*lr*), (3) Number of training epochs (*ep*), (4) Training batch size (*bs*), (5) Weight decay for the AdamW optimizer (*wd*). The search space ranges and further characteristics for each hyperparameter are summarized in Table 12.1.

Optimization Metric To address our research goal and optimize model performance for label suggestion in the ORKG (supporting knowledge graph population), we selected the macro-averaged F_1 score as the primary metric. This choice reflects the importance of balancing Precision (minimizing irrelevant label suggestions) and Recall (maximizing

Table 12.1.: Hyperparameter search space of selected parameters

Hyperparameter	Search Space Range	Data Type	Range Sampling
Seed	[1,000, 1,000,000,000]	integer	linear
Learning rate (*lr*)	[1e-6, 1e-4]	float	log
Number of training epochs (*ep*)	[10, 100]	integer	linear
Training batch size (*bs*)	[2, 4, 8, 16, 32]	integer	step
Weight decay (*wd*)	[1e-6, 1e-4]	float	log

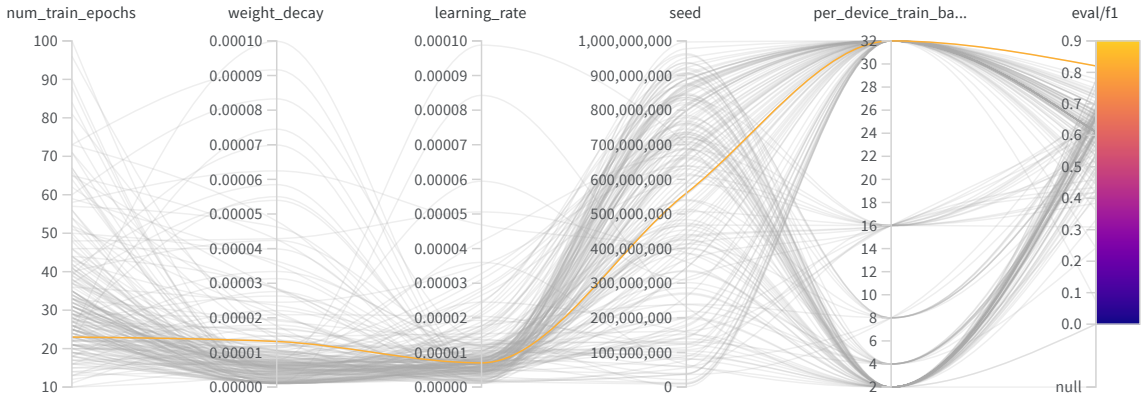

Figure 12.3.: DeBERTa hyperparameter search for *Property* with the augmented dataset

retrieval of relevant labels). Given the multi-label and imbalanced nature of the dataset (cf. Section 10.4.2), macro averaging ensures equal weighting across all classes, making it suitable for evaluating per-class performance regardless of label frequency.

Results of Hyperparameter Search To improve classification performance across the label categories Research Object, Evaluation Method, and Property, we conducted an extensive hyperparameter optimization using the aforementioned five transformer-based models. All models were optimized w.r.t. the macro-averaged F_1 score, which was selected due to its robustness in handling class imbalance by equally weighting performance across all classes (see above). Table A.2 shows the results of the hyperparameter search. The table summarizes the results of the hyperparameter search, organized by language model (LM), label category, and dataset variant (basic or augmented). Each search consisted of 200 trials, ensuring a comprehensive exploration of the hyperparameter space.

Hyperparameter sweeps were executed using the W&B framework, enabling visualization through parallel coordinates plots (see the figures in Section A.3.2 and Figure 12.3 for the category Property as an example). For each label category, we identified the best-performing configuration.

We conclude with the following key observations: *Data augmentation* considerably improved performance for underrepresented label categories (Research Object and Evaluation Method), with 80% of searches yielding higher F_1 scores. In contrast, the Property category showed improvement in only 40% of cases, likely due to its more balanced label

distribution. Training epochs ranged from 10 to 100. *Model efficiency* varied, i.e., BERT required ~20 fewer epochs on average compared to other models. Augmented datasets reduced training epochs from ~54 (basic dataset) to ~33, indicating faster convergence. *Learning rate* values were broadly distributed, with a concentration between 1.e-06 and 5.e-06, suggesting stable convergence at lower rates. *Weight decay* values between 1.e-05 and 5.e-05 yielded the best performance, with no clear dependency on model type or dataset variant. *Seed* values tended toward higher magnitudes, mitigating bias and enhancing reproducibility. *Batch size* selection showed no consistent pattern across label categories or datasets, though BERT-based models tended to utilize larger batch sizes than XLNet, potentially due to architectural differences.

12.4.1.2. Settings for Prompting

Following, we present the evaluation settings for prompting with a zero-shot approach.

Selection of Generative Language Models (LM) For the zero-shot approach, we use different language models: GPT-3.5 Turbo [206], GPT-4-0125-preview [207], LLaMA2:7B / 13B / 70B [288], Mistral:7B [122], and Mixtral:8x7B [123].

Hyperparameter and Prompt Optimization To conduct the experiments for the zero-shot classification approach, no hyperparameter tuning was required. To enhance reproducibility, we set the temperature parameter to zero ($temp = 0$), ensuring near-deterministic outputs across experimental runs. The experiments were executed using the OpenAI API and the API of an Ollama instance. For formulating the prompt, we used existing guidelines [227].

12.4.2. Evaluation Results

In this section, we present the empirical results of our evaluation, which is embedded in our automated evaluation strategy regarding the goal *applicability* of our proposed evaluation approach (cf. Section 5.3), across two main objectives: evaluating (i) classification performance and (ii) combined sustainability aspects (performance / energy consumption).

We present the results for the model training approach in Section 12.4.2.1, the results for the zero-shot approach in Section 12.4.2.2, and the evaluation of the environmental impact in Section 12.4.2.3. For the evaluation, we consider the metrics Precision (P), Recall (R), F_1 score (F_1), Jaccard, Hamming Loss (HL), and Specificity (Spec). All metrics have been macro-averaged.

12.4.2.1. Performance Results for Model Training Approach

In the following paragraphs, we analyze the performance of the trained language models for each category (i.e., *Research Object*, *Evaluation Method*, and *Property*). We begin by examining the results obtained using the basic dataset, followed by an evaluation of the extended dataset, which incorporates oversampling and data augmentation techniques. Thereby, we address our validation question VQ II-2.1.

Table 12.2.: Evaluation results of trained models for the *Research Object* category using the basic dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL	Spec
BERT	hp search	0.091	**0.692**	0.159	0.091	0.877	0.043
	test	0.090	**0.917**	0.151	0.083	0.877	0.059
	author	0.091	0.612	0.144	0.084	0.862	0.073
DeBERTa	hp search	0.256	0.269	**0.244**	**0.218**	**0.062**	0.983
	test	0.216	0.157	0.177	0.118	0.101	0.969
	author	0.090	**0.692**	**0.154**	0.090	0.910	0.000
DistilBERT	hp search	0.148	0.615	0.201	0.148	0.692	0.248
	test	0.095	0.821	0.164	0.093	0.697	0.238
	author	0.091	0.571	0.151	0.089	0.724	0.206
SciBERT	hp search	0.218	0.269	0.218	0.179	0.077	0.974
	test	0.403	0.228	0.277	0.191	0.084	0.979
	author	0.110	0.078	0.083	0.052	**0.093**	**0.981**
XLNet	hp search	**0.269**	0.231	**0.244**	**0.218**	**0.062**	**0.990**
	test	**0.474**	0.237	**0.306**	**0.221**	**0.077**	**0.984**
	author	**0.134**	0.104	0.099	0.063	0.103	0.955

Performance Results of Basic Dataset Table 12.2 presents macro-averaged performance metrics for various experimental runs assessing the **Research Object** category. The evaluation reveals substantial differences in performance across architectures and run types. XLNet consistently achieves the highest overall performance, with an F_1 score of 30.6%, Precision of 47.4%, and Jaccard of 22.1% on the test set. These results indicate strong predictive capability. SciBERT follows with an F_1 score of 27.7% and Precision of 40.3%, offering a balanced trade-off between Precision and Recall, though slightly lower than XLNet. DeBERTa performs well during hyperparameter search (F_1 = 24.4%) but drops to 17.7% on the test set, suggesting sensitivity to training conditions or potential overfitting. BERT and DistilBERT exhibited very high Recall (up to 91.7% for BERT) but extremely low Precision and F_1 scores (around 15.1% (for author run) to 16.4% (for test run)), indicating a tendency to overpredict labels. Specificity was highest for XLNet (98.4%) and SciBERT (97.9%), confirming their superior ability to avoid false positives. Overall, **XLNet** demonstrated the most robust and consistent performance, making it the most suitable model for this classification task. SciBERT represents a strong alternative, while BERT and DistilBERT may be considered when Recall is prioritized over Precision.

Table 12.3 presents macro-averaged performance metrics for the **Evaluation Method** category. The evaluation results reveal generally low performance across all models, reflecting the complexity of predicting evaluation methods from abstracts. Among the tested models, DistilBERT achieved the highest F_1 score on the test split (13.8%) with Precision (9.3%) and Recall (46.2%), indicating a moderate trade-off between Sensitivity and Specificity. BERT performed similarly, with an F_1 score of 12.5% on the test split, but exhibited higher Recall (50.0%) and lower Precision (8.2%), suggesting a tendency to overpredict labels. DeBERTa demonstrated strong Recall across all splits (up to 73.1%

Table 12.3.: Evaluation results of trained models for the *Evaluation Method* category using the basic dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL	Spec
BERT	hp search	0.136	0.308	**0.167**	**0.136**	0.385	0.566
	test	0.082	0.500	0.125	0.078	0.480	0.480
	author split	0.072	0.267	0.106	0.068	0.655	0.292
DeBERTa	hp search	0.105	**0.385**	0.155	0.105	0.756	0.187
	test	0.081	**0.731**	0.134	0.081	0.730	0.201
	author split	**0.095**	0.346	**0.129**	0.086	0.801	0.144
DistilBERT	hp search	0.096	0.308	0.139	0.096	0.397	0.554
	test	**0.093**	0.462	**0.138**	**0.086**	0.403	0.556
	author split	0.080	**0.462**	0.123	**0.080**	0.920	0.000
SciBERT	hp search	0.081	**0.385**	0.128	0.081	0.609	0.333
	test	0.069	0.538	0.112	0.069	0.618	0.333
	author split	0.074	0.378	0.113	0.073	0.647	0.288
XLNet	hp search	**0.138**	0.159	0.146	0.122	**0.083**	**0.941**
	test	0.090	0.136	0.106	0.072	**0.107**	**0.938**
	author split	0.070	0.070	0.066	0.042	**0.090**	**0.942**

on the test set) but low Precision, resulting in modest F_1 scores (13.4% on the test split). This pattern indicates that DeBERTa is effective at identifying relevant labels but lacks discrimination, leading to more false positives. SciBERT showed consistent but lower performance, with F_1 scores around 11.2% on the test split and balanced Precision-Recall ratios. XLNet, while achieving the highest Precision during hyperparameter search (13.8%), performed poorly overall, with the lowest F_1 scores on the test and author splits (10.6% and 6.6%), despite excellent Specificity (93.8%). Models such as DeBERTa and DistilBERT retained relatively better Recall, while XLNet maintained high Specificity but at the cost of Recall and F_1 score performance. Overall, **DistilBERT** offers the most balanced performance for this task, while DeBERTa excels in Recall but suffers from low Precision. XLNet is highly specific but underperforms in overall predictive capability. These results underscore the difficulty of evaluation method classification and the need for strategies to improve generalization and precision.

Table 12.4 presents macro-averaged performance metrics for various experimental runs assessing the **Property** category. The evaluation results show substantially higher performance compared to other categories, indicating that property-related labels are easier to predict from abstracts. BERT achieves the best overall performance on the test split, with an F_1 score of 59.4%, Precision of 62.9%, and Recall of 60.6%, demonstrating a strong balance between Precision and Recall. XLNet follows with an F_1 score of 54.8%, while SciBERT and DeBERTa performed similarly (F_1 approx. 53.1% – 53.5%). DistilBERT lags behind with an F_1 score of 45.7% on the test set. During hyperparameter search, DistilBERT and DeBERTa achieved the highest F_1 scores (78.5% and 73.6%, respectively), suggesting strong potential under optimized conditions. However, their performance dropped significantly on the test split, indicating possible overfitting or sensitivity to training configurations. In contrast,

Table 12.4.: Evaluation results of trained models for the *Property* category using the basic dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL.	Spec
BERT	hp search	0.708	0.508	0.579	0.413	0.308	0.717
	test	**0.629**	**0.606**	**0.594**	**0.435**	0.301	0.587
	author split	0.397	0.468	0.428	0.358	0.264	**0.637**
DeBERTa	hp search	0.875	0.658	0.736	0.593	**0.154**	**0.892**
	test	0.536	0.538	0.535	0.396	0.312	0.652
	author split	0.399	0.569	0.462	0.375	0.264	0.567
DistilBERT	hp search	**0.889**	**0.767**	**0.785**	**0.656**	**0.154**	0.733
	test	0.483	0.472	0.457	0.363	**0.258**	**0.655**
	author split	0.425	**1.000**	0.541	0.425	0.575	0.000
SciBERT	hp search	0.822	**0.767**	0.756	0.611	0.179	0.692
	test	0.537	0.556	0.531	0.394	0.301	0.556
	author split	0.435	**1.000**	**0.555**	**0.435**	0.517	0.067
XLNet	hp search	0.764	0.750	0.714	0.557	0.231	0.675
	test	0.605	0.533	0.548	0.384	0.344	0.587
	author split	**0.686**	0.449	0.485	0.363	**0.241**	0.627

BERT maintained consistent performance across runs, reinforcing its robustness. The author-based split introduced a notable decline in performance for all models. While Recall remained high for some models (e.g., DistilBERT and SciBERT reached 100%), Precision dropped sharply, leading to inflated Recall without meaningful improvements in F_1 scores. This pattern suggests that models tend to overpredict when faced with unfamiliar writing styles. Overall, **BERT** offers the most reliable and balanced performance for property classification, particularly on the test split. XLNet and SciBERT are strong alternatives, while DistilBERT and DeBERTa show high potential but require strategies to mitigate overfitting and improve generalization.

Performance Results of Augmented Dataset Table 12.5 presents macro-averaged performance metrics for the **Research Object** label category across three evaluation splits. The augmented dataset improves overall performance compared to the basic dataset, but results remain modest. XLNet achieves the highest scores during hyperparameter search, with an F_1 score of 33.3%, Precision of 34.6%, and Jaccard of 30.8%, indicating strong predictive capability and balanced Precision-Recall trade-off. On the test split, XLNet maintains superiority with an F_1 score of 24.5% and Precision of 30.3%, though Recall drops to 24.4%. Specificity remains high across all splits (>96.0%), confirming XLNet's ability to avoid false positives. DistilBERT performs competitively during hyperparameter search (F_1 score = 30.5%), but its test performance declines to 17.5%, suggesting sensitivity to unseen data. DeBERTa shows moderate results, with an F_1 score of 26.7% during hyperparameter search but only 19.1% on the test split. BERT and SciBERT exhibit high Recall (up to 92.3% for SciBERT on the test split) but very low Precision and F_1 scores, indicating a tendency to overpredict labels. This imbalance persists in the author-based split, where SciBERT

Table 12.5.: Evaluation results of trained models for the *Research Object* category using the augmented dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL	Spec
BERT	hp search	0.117	**0.615**	0.191	0.117	0.654	0.297
	test	0.088	0.692	0.149	0.083	0.695	0.260
	author split	0.085	**0.615**	0.148	0.085	0.740	0.184
DeBERTa	hp search	0.231	0.346	0.267	0.218	0.085	0.947
	test	0.247	0.196	0.191	0.126	0.115	0.950
	author split	0.151	0.106	0.112	0.069	0.114	0.962
DistilBERT	hp search	0.282	0.346	0.305	0.269	0.062	0.972
	test	0.205	0.157	0.175	0.122	**0.099**	**0.972**
	author split	0.067	0.054	0.059	0.035	0.111	0.971
SciBERT	hp search	0.139	0.538	0.186	0.139	0.600	0.362
	test	0.089	**0.923**	0.162	0.089	0.792	0.139
	author split	0.122	0.609	**0.189**	**0.111**	0.731	0.210
XLNet	hp search	**0.346**	0.346	**0.333**	**0.308**	**0.046**	**0.991**
	test	**0.303**	0.244	**0.245**	**0.179**	0.100	0.961
	author split	**0.173**	0.061	0.088	0.050	**0.096**	**0.986**

achieves the highest F_1 score (18.9%) but still struggles with Precision. Across all models, XLNet retains relatively better Precision and Specificity, while BERT and SciBERT prioritize Recall at the expense of Precision. Overall, **XLNet** remains the most effective model for this task under augmented conditions, offering the best balance between Precision, Recall, and overall F_1 score. DistilBERT and DeBERTa show potential but require strategies to improve robustness. High Recall from BERT and SciBERT suggests utility in Recall-critical scenarios, though their low Precision limits practical applicability.

Table 12.6 presents micro-averaged performance metrics for the **Evaluation Method** label category. Performance for the Evaluation Method category remains generally low, even after data augmentation, indicating that this classification task is inherently challenging. XLNet achieves the best overall performance during hyperparameter search, with an F_1 score of 26.0%, Precision of 28.6%, and Jaccard of 21.7%, suggesting a relatively balanced trade-off between Precision and Recall. On the test split, XLNet maintains competitive performance (F_1 = 13.9%), outperforming most other models in Precision (12.8%) and Jaccard (11.5%), while preserving high Specificity (93.1%). BERT shows the highest F_1 score on the test split (16.2%) due to its strong Recall (80.8%), but its Precision remains low (9.2%), indicating a tendency to overpredict. DistilBERT demonstrates similar behavior, achieving high Recall (83.1%) but modest Precision (7.3%), resulting in an F_1 score of 13.3%. These models prioritize recall over discrimination, which may be useful in recall-critical scenarios but limits overall accuracy. DeBERTa performs well during hyperparameter search (F_1 = 20.4%) but drops sharply on the test split (F_1 = 7.8%), suggesting overfitting or poor generalization. SciBERT exhibits the weakest performance on the test split (F_1 = 3.3%), despite achieving near-perfect Specificity during hyperparameter search, indicating an overly conservative prediction strategy. Across all models, the author-based split intro-

Table 12.6.: Evaluation results of trained models for the *Evaluation Method* category using the augmented dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL	Spec
BERT	hp search	0.087	0.385	0.137	0.087	0.715	0.240
	test	0.092	0.808	**0.162**	0.092	0.720	0.222
	author split	0.066	0.385	0.112	0.066	0.736	0.211
DeBERTa	hp search	0.190	0.236	0.204	0.184	0.100	0.924
	test	0.125	0.099	0.078	0.047	**0.110**	**0.949**
	author split	0.044	0.077	0.049	0.030	0.103	0.953
DistilBERT	hp search	0.119	**0.436**	0.165	0.106	0.854	0.086
	test	0.073	**0.831**	0.133	0.073	0.872	0.059
	author split	0.078	**0.462**	**0.134**	**0.078**	0.922	0.000
SciBERT	hp search	0.138	0.138	0.138	0.128	**0.058**	**0.991**
	test	0.024	0.057	0.033	0.020	0.118	0.944
	author split	**0.098**	0.078	0.079	0.050	**0.082**	**0.979**
XLNet	hp search	**0.286**	0.262	**0.260**	**0.217**	0.092	0.935
	test	**0.128**	0.160	0.139	**0.115**	0.123	0.931
	author split	0.071	0.112	0.085	0.060	0.106	0.945

duces further performance degradation, confirming that generalization to unseen authors remains a major challenge. DistilBERT retains relatively better Recall (46.2%) in this split, while XLNet continues to offer the highest Specificity (>94.0%). Overall, **XLNet** provides the most balanced performance under augmented conditions, particularly in Precision and Specificity, while BERT and DistilBERT excel in Recall but suffer from low Precision. DeBERTa and SciBERT show instability across splits, highlighting the need for improved generalization strategies.

Table 12.7 presents micro-averaged performance metrics for the **Property** label category. Performance for the Property category improves significantly with data augmentation, and all models achieve higher F_1-scores compared to the basic dataset. DeBERTa delivers the strongest overall performance, with an F_1-score of 82.0% during hyperparameter search and 71.1% on the test split, supported by high Precision (68.4%) and Recall (75.8%). Its Jaccard (56.8%) and low Hamming loss (26.7%) confirm robust predictive capability and balanced label coverage. BERT also performs well, reaching an F_1-score of 68.4% on the test split, driven by very high Recall (92.4%), though Precision (55.5%) is lower than DeBERTa. This indicates that BERT prioritizes Sensitivity, which may be beneficial in recall-critical scenarios. XLNet achieves competitive results, with an F_1-score of 64.6% on the test split and the highest Precision in the author-based split (65.4%), suggesting strong discrimination even under domain shifts. DistilBERT shows moderate performance, with an F_1-score of 62.7% on the test split, while SciBERT lags behind (F_1 = 49.3%), despite strong Precision during hyperparameter search. Across all models, the author-based split introduces performance degradation, particularly for BERT and DeBERTa, where Specificity drops sharply. XLNet demonstrates the best resilience under this split, maintaining an F_1-score of 62.5% and high Specificity (71.4%). Overall, **DeBERTa** emerges as the most effective

Table 12.7.: Evaluation results of trained models for the *Property* category using the augmented dataset

Language Model	Run Type	Precision	Recall	F_1	Jaccard	HL	Spec
BERT	hp search	0.571	0.833	0.647	0.482	0.386	0.424
	test	0.555	**0.924**	0.684	0.522	0.387	0.369
	author split	0.364	0.888	0.516	0.352	0.629	0.037
DeBERTa	hp search	**0.805**	**0.880**	**0.820**	**0.698**	**0.175**	**0.782**
	test	**0.684**	0.758	**0.711**	**0.568**	**0.267**	**0.714**
	author split	0.384	**1.000**	0.552	0.384	0.616	0.000
DistilBERT	hp search	0.557	0.847	0.649	0.505	0.456	0.303
	test	0.572	0.788	0.627	0.467	0.456	0.346
	author split	0.384	**1.000**	0.552	0.384	0.616	0.000
SciBERT	hp search	0.768	0.727	0.692	0.532	0.298	0.694
	test	0.543	0.561	0.493	0.337	0.453	0.536
	author split	0.599	0.576	0.522	0.371	0.371	0.607
XLNet	hp search	0.765	0.718	0.654	0.499	0.333	0.633
	test	0.651	0.742	0.646	0.479	0.347	0.583
	author split	**0.654**	0.691	**0.625**	**0.481**	**0.277**	**0.714**

model for property classification under augmented conditions, offering the best balance of Precision, Recall, and overall F_1-score. BERT excels in Recall, while XLNet provides strong Precision and robustness to author variation. DistilBERT and SciBERT remain viable alternatives but show less stability across splits.

Summary VQ II-2.1 (Results of the Fine-Tuned Classification Approach):
For evaluation, we investigated the effectiveness of transformer-based language models in predicting labels across the three label categories (Research Object, Evaluation Method, and high-level Property). A key challenge identified is the low performance for labels with few instances, which negatively impacts the macro-averaged F_1 score. The evaluation was conducted using a test dataset that remained independent of the training and hyperparameter tuning phases to ensure unbiased performance assessment. Results show that no single language model consistently outperforms others across all categories. On the basic dataset, XLNet achieves the highest macro F_1 score for the Research Object category, DistilBERT performs best for Evaluation Method, and BERT leads in the Property category. When oversampling and data augmentation techniques were applied, performance improved notably in the Property category, with DeBERTa achieving the highest F_1 score (from 53.5% to 71.1%) in the test set. These findings highlight the complexity of multi-label classification in domains with uneven label distributions and suggest that model selection and data augmentation strategies have to be tailored to specific label characteristics.

12.4.2.2. Performance Results for Prompting Approach

For the experiments involving trained language models, the dataset was partitioned into training, validation, and test subsets. Due to the limited number of data points in the dataset, we additionally employed a zero-shot classification approach by adopting a prompt-based strategy to improve generalizability, as discussed in the challenges in Section 12.2. To obtain label predictions without model-specific training, we designed a prompt that instructed the language model to assign labels based on a given label description and the abstract of each paper. The language models returned a list of predicted labels, which were then used to compute the performance metrics. Thereby, we address our validation question VQ II-2.1. The following results are based on the basic dataset.

Table 12.8 depicts the results of the zero-shot classification approach considering the *Research Object*, *Evaluation Method*, and *Property* categories.

For the category **Research Object**, GPT-4 achieves the highest scores across all metrics except Recall; LLaMA2:13B reaches 81.9%, but very low Precision and F_1 score, indicating a tendency to predict many labels with limited accuracy. Among the open-weights models, Mixtral:8x7B performs best, achieving a macro F_1 score of 27.7%. Overall, **GPT-4** demonstrates superior zero-shot classification capabilities for this task, while other models show trade-offs between Precision and Recall, often struggling with label accuracy and overprediction.

For the category **Evaluation Method**, GPT-4 outperforms all other models across most metrics, except Recall, where LLaMA2:13B again achieves the highest score (88.8%). However, this model also shows the lowest Specificity, indicating a tendency to overpredict labels, which explains its high Recall. Among the open-weights models, Mistral and Mixtral models perform moderately well, with F_1 scores of 30.8% and 28.6%, respectively, and reasonable Precision and Specificity. However, they do not match the performance of GPT-based models. Overall, **GPT-4** shows the most effective zero-shot classification capabilities for this category, while other models struggle with Precision and label discrimination, particularly in cases of high Recall and low Specificity.

For the category **Property**, LLaMA2:7B achieves the highest macro F_1 score of 56.75%, driven primarily by high Recall (98.6%). However, its Precision (42.6%) and Specificity (0.9%) are low, suggesting substantial overprediction and poor discrimination between relevant and irrelevant labels. The Hamming loss further indicates that LLaMA2:7B's predictions deviate more from the ground truth compared to other models, further highlighting the trade-off between Recall and overall prediction accuracy. GPT-3.5-turbo-0125 demonstrates the best Precision (45.6%) and a balanced Recall (69.9%), resulting in a competitive F_1 score (51.6%). GPT-4-0125-preview shows slightly lower performance across most metrics, with an F_1 score of 43.8% and a Specificity of 47.0%, the highest among all models, indicating better control over false positives. Mixtral:8x7B and Mistral:7B perform moderately, with F_1 scores of 51.4% and 49.9%, respectively, and balanced precision-recall trade-offs. LLaMA2:13B and LLaMA2:70B achieve high Recall but suffer from low Specificity and Precision, similar to LLaMA2:7B. Overall, while **LLaMA2:7B** excels in Recall, **GPT-3.5-turbo-0125** offers the most balanced performance across Precision, Recall, and Specificity, making it the most reliable model for zero-shot classification in the Property category.

Table 12.8.: Evaluation results of the zero-shot classification approach for different large language models

Language Model	F_1	Precision	Recall	Jaccard	HL	Spec
Research Object						
GPT-3.5-turbo-0125	0.306	0.243	0.569	0.194	0.173	0.846
GPT-4-0125-preview	**0.367**	**0.330**	0.535	**0.239**	**0.131**	**0.888**
LLaMA2:7B	0.148	0.085	0.686	0.083	0.574	0.377
LLaMA2:13B	0.145	0.082	**0.819**	0.081	0.725	0.217
LLaMA2:70B	0.182	0.108	0.783	0.106	0.546	0.417
Mistral:7B	0.238	0.221	0.510	0.142	0.230	0.788
Mixtral:8x7B	0.277	0.230	0.503	0.173	0.192	0.827
Evaluation Method						
GPT-3.5-turbo-0125	0.340	0.372	0.439	0.242	0.153	0.859
GPT-4-0125-preview	**0.397**	**0.413**	0.502	**0.284**	**0.128**	**0.869**
LLaMA2:7B	0.146	0.090	0.556	0.087	0.532	0.425
LLaMA2:13B	0.144	0.089	**0.888**	0.084	0.751	0.193
LLaMA2:70B	0.187	0.119	0.692	0.110	0.416	0.557
Mistral:7B	0.308	0.340	0.487	0.208	0.171	0.841
Mixtral:8x7B	0.286	0.284	0.477	0.188	0.172	0.839
Property						
GPT-3.5-turbo-0125	0.516	**0.456**	0.699	0.371	0.473	0.355
GPT-4-0125-preview	0.438	0.429	0.595	0.305	0.499	**0.470**
LLaMA2:7B	**0.568**	0.426	**0.986**	**0.424**	0.571	0.009
LLaMA2:13B	0.567	0.433	0.953	0.420	0.547	0.080
LLaMA2:70B	0.549	0.434	0.785	0.409	0.449	0.237
Mistral:7B	0.499	0.432	0.612	0.354	0.444	0.416
Mixtral:8x7B	0.515	0.452	0.671	0.386	**0.390**	0.348

Summary VQ II-2.1 (Results of the Zero-Shot Classification Approach):
For evaluation, we investigated the effectiveness of zero-shot classification using LLMs across our three label categories (Research Object, Evaluation Method, and high-level Property). GPT-4 achieves the highest macro-averaged F_1 scores for Research Object and Evaluation Method, while LLaMA2:7B performs best for Property, though its tendency to overpredict labels influenced the results. Compared to trained models, the zero-shot approach improves performance in the Research Object and Evaluation Method categories, but shows a decline in the Property, likely due to its more balanced label distribution favoring supervised learning. The zero-shot approach proves more effective for labels with few representatives, particularly in the Research Object category.
An observed anomaly with LLaMA2:13B revealed hallucinated labels that were provided during inference, suggesting that the model leveraged prior knowledge from pretraining. These findings indicate that zero-shot classification can be beneficial in imbalanced label scenarios, but its reliability depends on label distribution and model behavior.

12.4.2.3. Results for Environmental Impact

In the following paragraphs, we analyze the performance and energy consumption of the fine-tuned models and the zero-shot classification approach. Thereby, we address our validation question VQ II-2.2 using our metrics presented in Section 5.3.2.2. In the appendix, we provide detailed information about the carbon emission results (see Section A.3.3). Across all trained models, emissions vary slightly depending on the evaluation dimensions, but the overall trend remains consistent in this experimental setting: hyperparameter search dominates the carbon emissions compared to test runs. In the following, we concentrate on the evaluation results for the category *Property* in the test run.

Results of the Fine-Tuned Classification Approach The following analysis quantifies the environmental impact of transformer-based models across the inference phases using the basic and augmented datasets, while assessing the category *Property*. The results with the combined metrics for the test runs with the basic and augmented datasets are depicted in Table 12.9.

Results for combined metrics in the category* Property *(test run / basic).

BERT achieves the highest F_1 score (59.4%) while also exhibiting the lowest CO_2 emissions (2.413 g), resulting in the most favorable relative carbon efficiency (CE_{rel} = 4.06) and the highest Δ_{CE} (0.3204). In contrast, DeBERTa, despite achieving a comparable F_1 score (53.5%), incurred substantially higher emissions (13.315 g), leading to a considerably higher relative carbon cost (CE_{rel} = 24.87) and a considerably lower Δ_{CE} (0.0332). SciBERT and XLNet demonstrate intermediate performance and emissions, while DistilBERT, although more efficient in terms of emissions (5.666 g), yielded the lowest F_1 score (45.7%) and served as the baseline for Δ_{CE}. These results indicate that superior predictive performance does not necessarily entail increased environmental cost. Notably, BERT outperforms other models both in performance and carbon efficiency, suggesting that high performance can be achieved without compromising sustainability. Conversely, a marginal deviation in F_1 score, as seen with DeBERTa, may come at a disproportionate environmental cost.

Table 12.9.: Evaluation results of the model training classification approach, considering combined sustainability metrics for the *Property* category

Language Model	F_1	CO_2 (g)	n(CE)	CE_{rel}	Δ_{CE}
Property (test, basic dataset)					
BERT	**0.594**	**2.413**	**1.000**	**4.06**	**0.3204**
DeBERTa	0.535	13.315	0.000	24.87	0.0332
DistilBERT	0.457	5.666	0.702	12.38	*base*
SciBERT	0.531	8.986	0.397	16.91	0.0465
XLNET	0.548	4.480	0.810	8.17	0.1148
Property (test, augmented dataset)					
BERT	0.684	2.281	0.967	3.34	**0.5053**
DeBERTa	**0.711**	5.076	0.721	7.14	0.2596
DistilBERT	0.627	**1.908**	**1.000**	**3.04**	0.4241
SciBERT	0.493	6.063	0.635	12.29	*base*
XLNET	0.646	13.277	0.000	20.54	0.0698

Results for combined metrics in the category Property (test run / augmented) DeBERTa achieves the highest F_1 score (71.1%), indicating superior predictive performance. However, this comes with moderate CO_2 emissions (5.076 g), resulting in a relative carbon efficiency (CE_{rel}) of 7.14 and a Δ_{CE} of 0.2596. BERT, while slightly trailing in F_1 score (68.4%), demonstrates better environmental performance, emitting only 2.281 g of CO_2 and achieving the highest Δ_{CE} (0.5053) among all models. DistilBERT offers a favorable trade-off between performance and sustainability. It attains a competitive F_1 score of 62.7% with the lowest CO_2 emissions (1.908 g), resulting in the best relative carbon efficiency ($CE_{rel} = 3.04$) and a strong Δ_{CE} (0.4241). In contrast, XLNet reaches an F_1 score of 64.6% but incurs the highest emissions (13.277 g), leading to the lowest normalized carbon efficiency ($n(\text{CE}) = 0.000$) and a low Δ_{CE} (0.0698). SciBERT records the lowest F_1 score (49.3%) and relatively high emissions (6.063 g), serving as the baseline for Δ_{CE}. These findings highlight that similar levels of predictive accuracy can be associated with vastly different environmental costs. Notably, models such as DistilBERT demonstrate that it is possible to achieve both relatively competitive performance and high carbon efficiency. Such models may be overlooked if environmental impact is not explicitly considered during model selection.

Results of the Zero-Shot Classification Approach The reported energy consumption and carbon emissions for GPT-based models in the appendix (Section A.3.3) underestimate the true environmental impact, as they exclude upstream computational costs such as model inference infrastructure and backend processing. For these models, only the energy consumption associated with API access can be recorded. As this causes a major threat to validity, we restrict the following analysis to the open-weight models. The results for the zero-shot classification approach with the combined metrics are depicted in Table 12.10.

Table 12.10.: Evaluation results of the zero-shot classification approach, considering combined sustainability metrics for the *Property* category

Language Model	F_1	CO_2 (g)	n(CE)	CE_{rel}	Δ_{CE}
LLaMA2:7B	**0.568**	15.898	0.984	28.01	**0.0470**
LLaMA2:13B	0.567	38.507	0.911	67.89	0.0193
LLaMA2:70B	0.549	321.501	0.000	585.93	0.0017
Mistral:7B	0.499	**10.937**	**1.000**	**21.91**	*base*
Mixtral:8x7B	0.515	25.980	0.952	50.50	0.0064

Results for combined metrics in the category (Property). Among the evaluated models in the zero-shot generative setting, LLaMA2:7B achieves the highest F_1 score (56.8%), while emitting 15.898 g of CO_2, achieving the highest Δ_{CE} in this run. Mistral:7B, although yielding the lowest F_1 score (49.9%), exhibits the most favorable carbon profile, with the lowest CO_2 emissions (10.937 g), the highest normalized carbon efficiency ($n(\text{CE}) = 1.00$), and serving as the baseline for Δ_{CE}. As model size increases, emissions rise steeply without proportional gains in performance. LLaMA2:70B, for instance, emits over 321.501 g of CO_2; more than 30 times higher than that of Mistral:7B, while achieving only a marginally higher F_1 score (54.9%). This results in the lowest normalized carbon efficiency ($n(\text{CE}) = 0.000$) and a minimal Δ_{CE} (0.0017). Similarly, Mixtral:8x7B and LLaMA2:13B show moderate to good performance but incur substantial environmental costs, with CO_2 emissions of 25.980 g and 38.507 g, respectively. These findings underscore the pronounced trade-off between model size (e.g., Mistral vs. Mixtral) and environmental impact in zero-shot generative settings. While larger models may offer slight improvements in predictive performance, they do so at a disproportionately high environmental cost. Consequently, smaller models like Mistral:7B may offer a more sustainable alternative, particularly when marginal performance gains are not critical.

> ***Summary VQ II-2.2 (Environmental Impact with Combined Metrics):***
> *We tracked energy consumption and carbon emissions across model training, hyperparameter optimization, and inference. Hyperparameter search was the most energy-intensive phase. Overall, the results highlight a trade-off between performance and sustainability with our combined metrics. Optimal model selection should balance performance with environmental impact. In comparison to larger models, smaller models often achieved competitive F_1 scores with considerably lower carbon emissions and better scores w.r.t. the combined metrics, suggesting that careful model selection can reduce environmental impact without sacrificing effectiveness.*

Conclusion In the context of Open Science and the FAIR principles, where transparency, accessibility, and sustainability are paramount, these results underscore the need for energy-aware model selection based on combined sustainability metrics (see Section 5.3.2.2), especially when AI-based classifiers are embedded in real-world, software-intensive systems.

12.5. Threats to Validity

In this section, we discuss potential threats to validity (TtV) of our experimental design and setup by referencing the TtV classes defined in our classification and data schema (see Table 7.1), and outline corresponding mitigation strategies.

External Validity refers to the generalizability of experimental results beyond the specific experimental context. In our experiments, we used a dataset derived from a literature study (see Section 10.4.2), which serves as a domain-representative gold standard for SWA research. The classification labels for evaluation methods are based on the ACM Empirical Standards [232], ensuring relevance across the broader SE domain. Additionally, the property category is aligned with the ISO/IEC 25010:2011 standard for software quality (SQuaRE) [114], further supporting applicability within SE research. However, the research object category was tailored to SWA research based on ICSA and ECSA papers (see Section 7.3). As a result, the dataset may not fully reflect the diversity of research artifacts or classification tasks across other subfields of SE. Despite this limitation, the underlying classification framework is designed to be generalizable to hierarchical schemas and adaptable across research domains. Future work should extend the evaluation to additional datasets and domains to further validate the generalizability of the approach.

Internal Validity concerns whether observed effects are attributable to the experimental treatment rather than confounding factors. In our experiments, we used our gold standard dataset for SWA research. For the classification task, we relied on paper abstracts as input, assuming they function as self-contained, standalone representations of the research work. To ensure consistency, we applied standardized preprocessing techniques and identical evaluation procedures across all models. However, there remains a threat that abstracts alone may not provide sufficient information for accurate classification, as indicated by our retrospective analysis of the abstracts (see Section 12.6).

Construct Validity refers to the extent to which the experimental setup accurately reflects the research objectives. To ensure construct validity, we followed established practices for prompt engineering, e.g., recommendations for prompt construction and fine-tuning, including stratified data splitting and hyperparameter optimization using Optuna. We employed standard performance metrics (e.g., Precision, Recall, and F_1 score) alongside our proposed sustainability metrics to evaluate trade-offs between model accuracy and resource efficiency (see Section 5.3.2.2). Nonetheless, the limited number of configurations (hyperparameter space) tested and the reliance on automated metrics may underrepresent the full performance landscape and introduce mono-method bias. These risks were mitigated by grounding our evaluation in widely accepted frameworks and metrics.

Confirmability and Repeatability are combined discussed as potential threats. Confirmability refers to the extent to which experimental findings are free from researcher bias and grounded in the data, while repeatability concerns the ability to reproduce results under similar conditions. All experimental artifacts, including datasets, code, and evaluation

scripts, are tracked with W&B and publicly available to support independent verification. We report both favorable and unfavorable research results. The inherent non-determinism of LLMs, particularly in prompting-based approaches, poses a threat to exact reproducibility. To address this, we used fixed random seeds for fine-tuning and set the temperature to zero for zero-shot prompting in each experiment. Our experimental setup is fully documented and publicly available, enabling replication with minimal variability. Our experimental setup's prerequisites have to be fulfilled and aligned with the best practices discussed in Section 10.2.

12.6. Retrospective Analysis

Since both automated classification strategies yielded suboptimal to moderate and good performance (see Section 12.4), we conducted a more in-depth analysis of the dataset quality, focusing on the information content of abstracts. To this end, we introduce a classification schema for structured abstracts in SE, designed to support Open Science principles and align with FAIR guidelines. Each class in the schema corresponds to a specific heading that maps to a sentence within the abstract, enabling a structured representation of a paper's content. In the following, we first present our research approach for deriving the classification schema for structured abstracts in SE (Section 12.6.1). Then, we introduce our corresponding classification schema in Section 12.6.2. Finally, we present key findings and experiences during the classification process of the SWA dataset abstracts, and conclude with implications for the research practice in writing abstracts (Section 12.6.3).

12.6.1. Background and Research Approach

Abstracts serve as concise, self-contained summaries that distill the core contributions and findings of scientific papers. In SE research, where research spans diverse paradigms, including empirical investigations, constructive solution proposals, and industry experience reports (cf. Chapter 7), abstracts play a critical role in enabling researchers to quickly assess the relevance and value of a publication and supporting dataset quality for an automated approach. While structured abstracts are widely adopted in fields such as medicine (e.g., IMRAD[3] comprising Introduction, Methods, Results, and Discussion), their use in SE remains limited and inconsistent. Existing approaches for structured abstracts in SE rely on those standards (see Section 14.2.3). SE publication venues vary in their requirements, with only a few mandating structured abstracts and most leaving the format to the discretion of authors. This heterogeneity in abstract structure hinders rapid comprehension, reduces clarity (cf. Budgen et al. [35] and Kitchenham et al. [149]), and complicates the retrieval and comparison of research contributions; challenges that are particularly pronounced in interdisciplinary and data-driven research contexts. To address these limitations, we propose a set of headings for structured abstracts in SE to label their sentences. This schema is designed to promote Open Science and FAIR principles and to capture domain-specific aspects for SE, encompassing both problem-oriented investigations and solution-oriented developments based on a paper's findings, validity, and evidence.

[3] `https://pmc.ncbi.nlm.nih.gov/articles/PMC442179/` [Last accessed on 2025-10-19]

Pre-Study To support Open Science and align with the FAIR principles, we developed a classification schema for structured abstracts tailored to the specific characteristics of SE research. Given the diversity of SE contributions, ranging from empirical investigations to constructive solution proposals and industry experience reports, our goal was to establish a standardized yet flexible structure that enhances transparency, discoverability, and reusability of research outputs. We adopted a bottom-up, data-driven approach to derive the classification. As a foundation, we selected a representative corpus of high-quality publications from top-tier SE venues, including the ICSE, ESEC/FSE, EASE, ESEM, EMSE, ECSA, and ICSA. These venues were chosen because of their methodological rigor, diversity of research contributions, and relevance to both academic and industrial SE research. The derivation process involved a sentence-level qualitative analysis of abstracts from the selected corpus. Initially, we defined a set of five core labels, namely *Introduction*, *Research Objective*, *Research Method*, *Results*, and *Findings*, based on common structural elements observed in scientific abstracts. These labels were iteratively refined and expanded as new patterns and content types emerged during the analysis by adding *Limitation*, *Replication Package*, *Focus*, and *Future Work*. This iterative coding process allowed us to capture the nuanced ways in which SE researchers report their work, ensuring that the classification reflects the dual nature of SE as a problem- and solution-oriented discipline and beyond. The resulting classification aims to support both human-readable and machine-processable representations of research abstracts. It enables automated annotation and segmentation of unstructured abstracts, thereby facilitating improved information retrieval, systematic reviews, and meta-analyses. Moreover, by promoting consistent reporting practices, the classification contributes to the broader goals of Open Science, including enhanced reproducibility, transparency, and data interoperability in SE research.

Classification Process Based on the finalized classification schema for structured abstracts in SE, comprising defined labels, their corresponding descriptions, and representative sentence examples (cf. Section 12.6.2), two reviewers, one of whom was the author of this thesis, independently annotated the gold standard dataset. This collaborative labeling process was guided by a shared understanding of the classification criteria beforehand, ensuring consistency and reliability in the application of labels across the dataset as best as possible. The dataset was cleaned, and the existing headings of structured abstracts were removed to avoid bias. Each sentence was labeled by the reviewers with a primary label indicating its main focus and intent. If necessary, an alternative label was also assigned. All labels were accompanied by a confidence score ranging from 1 (low confidence) to 5 (high confidence). The resulting classifications were discussed by the reviewers, deriving an experience report and implications for the SWA dataset (see Section 12.6.3).

12.6.2. Structured Abstracts in Software Engineering

In this section, we introduce our proposed set of structured abstract headings for SE research papers, along with their corresponding definitions. We conceptualize SE as a constructive research discipline encompassing two primary domains: the problem domain (focused on empirical investigation of SE challenges) and the solution domain (centered on the development of novel approaches and technologies). To support clarity,

consistency, and comparability in research reporting, we propose a structured abstract format comprising nine (partially optional) categories:

Introduction defines the broader context of the paper within SE, SE meta-research, or SE education and training. This heading introduces the research topic, outlines the relevant problem or solution domain, and states the research objective, question, or hypothesis. It also summarizes prior work to position the study within the current state of the art and identify existing research gaps.

Example: "Academic research, by its nature, is notorious for being a challenging and demanding field." [4]

Research Objective states the aim, purpose, or goal of the research and defines the steps to achieve it. This heading distinguishes, among other things, the following main research intents: (i) Empirical Problem: the empirical investigation of an SE problem; (ii) Constructive Approach: the development of novel solutions such as methods, models, meta-models, or algorithms; and (iii) Evaluative Investigation: the assessment of proposed solutions through defined evaluation goals.

Example: "In this paper, we present a tool-supported approach to estimate the change propagation caused by a change request in business processes or software systems based on the software architecture and the process design." [239]

Research Method describes how the research was conducted, including the study design, selected research method, research setting, and the type of analysis performed.

Example: "For the evaluation, we used two case studies." [191]

Results presents the outcomes derived from the applied research method and associated metrics. Results are categorized by research level: [primary], [secondary], and [tertiary]; and by the types of results: [quantitative], [qualitative], and [mixed].

Examples:

[primary] "The results also show a significant agreement that a lack of experience especially in academic writing and supervision problems constitute the most influential barriers to publishing." [4]

[secondary] "17 papers (11%) provide replication packages and 97 papers (63%) explicitly state threats to validity." [153]

Findings interpret, discuss, and contextualize the research *Results*.

Example: "In particular, this paper notes the relevance of group decision-making, decision scope, and social factors such as trust in effective architecture decision-making." [224]

Limitation refers to the [Limitations of the empirical study] to the corresponding [Research Method] and/or to the [Limitations of the solution approach]. Thus, *Limitation* addresses various shortcomings of a paper.

Example: "This approach has not been evaluated extensively, yet." [162]

Focus represents the target group of the paper, e.g., researchers or practitioners. This heading may be an optional item. Contributions / benefits for the focus group are stated.

Example: [Focus group: researchers (authors, reviewers)] "Researchers can use our results to find recommendations on relevant properties to evaluate and evaluation methods to use and to identify reusable evaluation artifacts to compare their novel ideas with other research." [153]

Future Work outlines recommendations for the next research steps and ongoing or future work in terms of engineering steps for a solution approach or further research designs and options.

Example: "In addition, the results motivate future research lines to support scalability (i.e., to deploy the architecture on a larger scale) and security." [159]

Replication Package states the availability and accessibility of supplementary research artifacts associated with the paper. We differentiate between [Empirical artifacts], which refer to artifacts related to the research method, and [Solution artifacts], which represent (software) artifacts associated with a new solution approach.

Example: [Empirical artifact] "We have made the dataset publicly available that will serve as a starting point for researchers to further reference and investigate the design decision detection and classification problem." [30];

Note: This example also mentions a focus group.

12.6.3. Experience, Results, and Implications

During the manual annotation of the abstract sentences in the SWA gold standard dataset, several practical issues during the labeling process were observed and discussed by the two reviewers.

First, it became evident that individual sentences often encapsulate multiple aspects of a study, making it difficult to assign a single, unambiguous label. This lack of separation of concerns within sentences, such as simultaneously describing both the research objective and the method, complicated the classification process and occasionally required interpretative judgment. Thus, the classification task resulted in a multi-class problem.

Second, the classification of certain sentences was highly dependent on their surrounding context. In particular, the meaning and function of a sentence could not always be determined in isolation, necessitating the consideration of preceding and subsequent sentences to ensure accurate labeling. This contextual dependency highlights the limitations of sentence-level classification when abstracts are not explicitly structured.

Third, distinguishing between specific categories, most notably *Results* and *Findings*, as well as *Research Objective* and *Research Method*, proved particularly challenging and the highest disagreements. These categories often overlap semantically in abstracts, especially when authors do not clearly delineate between the generation of results and their interpretation. Interestingly, abstracts that were already written in a structured format (e.g., using labeled sections) exhibited significantly higher clarity and inter-rater

agreement during the labeling process. This observation reinforces prior findings that structured abstracts improve comprehensibility and consistency in research reporting.

Fourth, a recurring issue was the lack of sufficient detail in many SE abstracts, which hindered the accurate assignment of labels. This was particularly evident in lower-quality abstracts, where essential information, such as the research method or evaluation details, was either missing or ambiguously formulated.

Finally, the distribution of labels was highly imbalanced. Notably, approximately three-quarters of all sentences were assigned to the main labels *Introduction* and *Research Objective*, indicating a predominant focus on motivating the research rather than presenting the paper's findings as a stand-alone entity. These findings emphasize the need for clearer and more balanced abstract-writing practices in SE and for the development of structured abstract guidelines tailored to the field's characteristics. They also demonstrate the potential of structured abstracts to improve clarity, enable automated classification, and advance Open Science and FAIR-compliant research dissemination.

12.7. Conclusion

This chapter addressed key limitations in manual KG population to speed up the task. In response, we introduced a classification framework that integrates two paradigms based on LLMs: transfer learning and prompt-based classification. This framework is specifically designed for multi-label classification problems to cope with the complex description of research artifacts with different classification concepts and knowledge-based descriptive metadata. As an advanced NLP tool, the framework incorporates several features in terms of, e.g., author-based dataset splitting, data augmentation strategies, hierarchical level-based classification, systematic hyperparameter optimization, and integrated experiment tracking. Thus, these current features address various challenges for an automated approach, as discussed in Section 12.2. We demonstrated the feasibility and maturity of our classification framework through evaluation via a use case example in SWA research using our dataset (see the dataset analysis in Section 10.4.2) derived from our literature study (see Chapter 7). The framework helps researchers and developers to construct, evaluate, and benchmark classifiers across evolving and hierarchical classification schemas. It addresses key challenges in automated KG population and supports tasks such as knowledge base construction and literature review updates. Moreover, the framework is extensible and adaptable to various real-world applications. It supports performance and sustainability trade-offs, including considerations of energy consumption and carbon emissions, thereby promoting responsible AI practices and fostering reuse within the research community.

13. A Retrieval Framework and Knowledge Access Module

This chapter builds on the publication "HubLink: A Novel Question Answering Retrieval Approach over Knowledge Graphs" [134] published at the 24th International Semantic Web Conference (ISWC2025), Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG). We ask the following research question along with its corresponding validation questions:

RQ II-3: How can the limitations of schema-dependent and training-based KGQA systems in scholarly domains be overcome by integrating RKGs with LLMs for schema-agnostic and provenance-aware retrieval?

VQ II-3.1: What is the retrieval performance of such an approach when benchmarked against state-of-the-art approaches on scholarly datasets?

VQ II-3.2: What is the answer generation performance of such an approach when benchmarked against state-of-the-art approaches on scholarly datasets?

This chapter introduces a novel retrieval approach, HubLink (cf. Section 13.2), embedded in a graph-based RAG-inspired pipeline within our retrieval framework (Section 13.3). By leveraging pre-trained LLMs, HubLink enhances scholarly search tasks with semantic depth and enables provenance-aware retrieval based on `RDF`-based knowledge graphs.

The retrieval framework supports configurable evaluations in a RAG-inspired system for benchmarking and the construction of domain-specific Question Answering (QA) systems.

For evaluation, we use the ORKG (Section 2.6) as the underlying knowledge base with four graph variants (see Figure A.1) and utilize the Software Architecture (SWA) research dataset (see Chapter 7) to populate the graph (see Section 11.3). We evaluate our approach regarding retrieval quality and answer generation quality (Section 13.4.2) by using traditional RAG metrics and LLM-as-a-Judge assessments (see Section 2.8.6) against three state-of-the-art baselines. All baselines are published in top-tier ACL (Association for Computational Linguistics) venues. The evaluation follows the extrinsic strategy of our evaluation approach (Section 5.3).

Consequently, this chapter constitutes the contribution C7.

13.1. Problem Statement

Despite major advances in digital publishing (see Section 3.2.2), scholarly communication remains predominantly document-centric [13, 117]. Scientific knowledge is typically

disseminated through content-isolated digital articles (PDF files), which hinders the interlinking of related methodologies, findings, and datasets. This poses substantial barriers to knowledge access and discovery, particularly in light of the exponential growth of scientific literature (cf. the data analysis of the number of published papers in SE per year by Garousi et al. [75]). Traditional keyword-based search methods further intensify the problem due to lexical variability, such as synonyms, abbreviations, and misspellings, making literature search a cognitively demanding and time-intensive task (cf. description of Problem Statement 2).

Recent developments in LLMs (based on the Transformer architecture, see Section 2.3.1) offer promising opportunities for natural language-based scholarly search. LLMs enable question answering with intuitive interfaces that allow researchers to query large corpora using conversational language. However, relying solely on the internal knowledge of pre-trained LLMs often results in shallow, unverifiable, or hallucinated responses [213]. To mitigate these limitations, RAG has been proposed, wherein LLMs are supplemented with context retrieved from external knowledge sources [180]. While RAG aims at improving the reliability of responses, naive implementations, typically based on embeddings and vector stores (e.g., ORKG Ask in Section 14.2.4), suffer from limited retrieval precision and poor synthesis of information across multiple documents [213] (see also Section 10.1).

To address these issues, structured representations of scholarly knowledge have gained attention, particularly in the form of RKGs, such as the Open Research Knowledge Graph (ORKG) [13, 117]. RKGs encode entities and relationships extracted from scientific content in a machine-readable format, enabling expressive and interconnected queries that transcend the limitations of document-centric retrieval (see Section 2.5.2.3).

The integration of LLMs with KGs for QA, referred to as KGQA, has shown success in open-domain applications [213]. However, its adaptation to the scholarly domain remains underexplored. Existing scholarly KGQA approaches predominantly rely on Semantic Parsing (SP), which translates natural language queries into structured queries (cf. Section 14.2.4). While effective under controlled conditions, SP methods are constrained by their dependence on manually curated training data and domain-specific query templates. These systems scale poorly across diverse research domains, struggle to adapt to evolving RKG schemas, and fail to capture the nuanced semantics of academic discourse. We identified a critical gap in scholarly KGQA research and, thus, proposed an alternative retrieval strategy that leverages LLMs and embeddings in a schema-agnostic, training-free manner.

13.2. HubLink: A Novel Retrieval Approach

In this section, we introduce HubLink, a novel schema-agnostic and training-free retrieval approach to KGQA, that employs a graph-based RAG-inspired pipeline comprising indexing (Section 13.2.1), retrieval (Section 13.2.2), and answer generation (Section 13.2.3) stages. Thereby, we address our research question:

RQ II-3: How can the limitations of schema-dependent and training-based KGQA systems in scholarly domains be overcome by integrating RKGs with LLMs for schema-agnostic and provenance-aware retrieval?

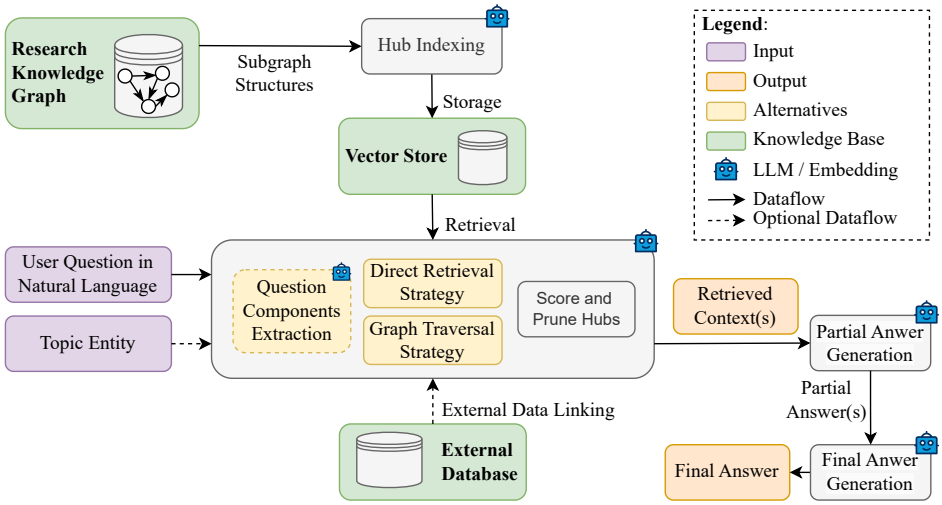

Figure 13.1.: Overview of our retrieval approach HubLink, covering the phases of indexing, retrieval, and answer generation (Figure adapted from Kaplan et al. [134])

HubLink partitions the knowledge graph into distinct substructures called *Hubs* during the indexing phase. These Hubs enable source-aware retrieval, allowing provenance tracking of retrieved information, which is a particularly valuable and mandatory feature in scholarly search contexts where the traceability of (literature) references is essential.

Figure 13.1 provides an overview of our retrieval approach.

13.2.1. Indexing

HubLink's indexing process transforms KGs into structured sets of interlinked subgraphs, referred to as *Hubs*, which serve as the primary retrieval units for question answering. This schema-agnostic approach begins by selecting start entities as initial traversal points. Through graph traversal, the approach identifies *Hub Root* entities that anchor directed paths to terminal nodes, forming the basis of each hub. Each hub consists of a hub root and its associated *Hub Paths*, which connect the root to leaf nodes or other hub roots. These paths are semantically coherent substructures that encapsulate meaningful segments of the graph. Once identified, each path undergoes semantic enrichment: it is processed by an LLM to generate a textual description, and its constituent triples, entities, and predicates are extracted. These elements are then embedded into a low-dimensional vector space using a pre-trained embedding model and stored in a vector database to enable efficient approximate nearest neighbor (ANN) search [316, 56] during retrieval. Metadata, including the hub identifier and LLM-generated descriptions, are attached to each vector to ensure traceability and support provenance-aware retrieval. The indexing process is recursive: it continues from the endpoints of each hub path unless a leaf node is reached or a predefined traversal depth is exceeded. This indexing strategy enables transparent and interpretable retrieval, aligning with the demands of scholarly search systems that require both semantic relevance and source traceability.

Index Update Since the HubLink retriever relies exclusively on the indexed content of the KG, maintaining index consistency is essential to ensure accurate and up-to-date responses. Any modification to the graph must be reflected in the index to preserve alignment between the data sources and retrieval results. To ensure a consistent KGQA system in dynamic graph environments, our approach incorporates two complementary update strategies: a *Fixed Update Strategy* for scheduled, full synchronization of the index, and a *Dynamic Update Strategy* that responds to real-time graph changes by selectively updating affected components. Together, these strategies provide a robust framework for ensuring that HubLink's index remains consistent with the evolving structure of the underlying and populated knowledge graph, thereby supporting reliable and provenance-aware retrieval.

13.2.2. Retrieval

In this section, we outline the retrieval mechanism employed by the HubLink approach, which identifies relevant hubs and associated paths to construct partial answers that are subsequently merged into a final response. Two retrieval strategies are introduced: the *Direct Retrieval Strategy* (Section 13.2.2.2) and the *Graph Traversal Strategy* (Section 13.2.2.3). Both strategies require question preprocessing, which captures the semantic structure and vector representations of the input query.

13.2.2.1. Question Processing

The retrieval process in HubLink begins with the semantic preprocessing of the input question. This step is essential for enabling precise and context-aware matching against indexed graph structures. The question is first embedded into a vector representation using a pre-trained embedding model. Subsequently, an LLM is employed to extract key semantic components, such as entities, types, temporal expressions, and contextual constraints that characterize the query. Each extracted component is individually embedded, and the resulting vectors are stored alongside the original question embedding. These representations collectively form a structured input, which includes the original query, its semantic decomposition, and the corresponding vector embeddings. This enriched representation allows the retriever to perform fine-grained matching during the retrieval phase, particularly for complex queries involving multiple constraints. By decomposing and embedding both the question and its components, HubLink enhances retrieval precision and supports transparent, provenance-aware question answering.

13.2.2.2. Direct Retrieval Strategy

The direct retrieval strategy in HubLink enables efficient semantic search over the indexed knowledge graph without accessing the graph structure during query execution. This approach relies entirely on approximate nearest neighbor (ANN) search [316, 56] within the vector store, making it suitable for fast retrieval scenarios. After the initial decomposition into key components, as described in the previous section, the approach uses these components to guide the retrieval of relevant hubs. Candidate hubs are iteratively collected until a predefined threshold is reached (`HubsToKeep`), ensuring diversity and relevance

through similarity-based filtering. To maintain consistency across retrieved hubs, a path normalization step ensures that each hub contains a limited number of semantically rich paths (`TopPathsToKeep`). If a hub exceeds the threshold, redundant paths are pruned; if it falls short, additional paths are retrieved based on relevance to the query. This guarantees balanced representation across hubs. Once the hubs are filtered and ranked, partial answers are generated for each hub. These are then aggregated into a final response. If no relevant information is found, the system returns an empty result. This strategy offers a scalable and responsive solution for KGQA, balancing retrieval speed with semantic precision by leveraging precomputed embeddings and structured query representations.

13.2.2.3. Graph Traversal Strategy

The graph traversal retrieval strategy in HubLink enables context-sensitive exploration of the knowledge graph by navigating its structure level by level. Unlike the direct retrieval strategy, this approach accesses the graph during query execution, allowing for more precise retrieval in localized contexts. The process begins by initializing a set of root entities for bidirectional traversal, both forward and backward, from the topic entity. The input question is then preprocessed to extract semantic components and vector embeddings (see Section 13.2.2.1). Retrieval proceeds iteratively across traversal levels, incrementing until the maximum depth is reached. At each level, the algorithm checks whether any entities remain to be explored. If none are found, the process terminates. Otherwise, the system invokes a retrieval function to identify candidate hubs and determine the next set of entities for traversal. These hubs are filtered and ranked based on semantic relevance, and partial answers are generated using the most relevant hubs. If partial answers are produced, they are aggregated into a final response; otherwise, the algorithm continues to the next level. The retrieval function identifies hub roots and collects up to `TopPathsToKeep` semantically relevant paths for each. If no paths are found for a hub, it is skipped. Otherwise, a new hub is instantiated and added to the result set. The function returns both the list of hubs and the next entities to explore. The graph traversal strategy thus offers a more precise but computationally intensive alternative to direct retrieval, making it suitable for complex queries requiring deeper semantic context.

13.2.2.4. Pruning Hubs

To improve retrieval relevance and efficiency, HubLink employs a pruning mechanism that filters out semantically irrelevant hubs. Each candidate hub is assigned a weighted score based on the relevance of its associated paths. The score is computed as a weighted average, where higher-scoring paths are emphasized using an exponential weighting schema.

Let $\vec{s} = [s_1, s_2, \ldots, s_n]$ be the list of path scores for a given hub. The weights are computed as $w_i = \exp(\alpha \cdot s_i)$, and the overall hub score is calculated as $\text{Score}_{\text{Hub}} = \frac{\sum_{i=1}^{n} w_i \cdot s_i}{\sum_{i=1}^{n} w_i}$.

This formulation allows highly relevant paths to dominate the hub score, while suppressing the influence of less informative ones. After scoring, hubs are ranked and the top `HubsToKeep` are selected for further processing, ensuring semantic precision and computational efficiency in the retrieval pipeline.

13.2.3. Answer Generation and Information Linking

After relevant hubs are retrieved, HubLink proceeds with answer generation, which follows a unified process across all retrieval strategies. Each hub can be enriched through a linking procedure, which supplements graph-derived content with external contextual knowledge (e.g., text from the full paper). This step enhances the semantic richness of sparse or incomplete graph segments. **Partial answers** are generated for each hub using a language model. These answers incorporate the original question, textual descriptions and triples from the hub paths, and any linked knowledge. If the graph traversal strategy is used, the traversal path from the topic entity is also included to provide relational context. A carefully designed prompt ensures that the language model produces informative partial answers only when sufficient evidence is available. Finally, partial answers are aggregated into a coherent final response. This modular approach enables flexible, provenance-aware question answering, even in complex or incomplete graph environments.

The final step in HubLink's retrieval pipeline involves synthesizing partial answers into a coherent response in the **Final Answer Generation**. This is achieved by prompting a language model with the original question, the set of partial answers, and the corresponding HubPath objects. The model integrates the evidence and insights from each hub to produce a unified answer. For literature search tasks, the output is formatted with citation markers (e.g., [i]) linked to a list of referenced sources, enhancing interpretability and traceability. To support explainability, HubLink also returns the set of triples that contributed to the final answer. These triples are selected by ranking the original HubPaths and filtering their triples using a language model, which identifies those directly relevant to the synthesized response. This ensures that the output is not only contextually accurate but also transparently grounded in the underlying graph data.

13.3. Retrieval Framework

In this section, we introduce our retrieval framework developed to evaluate the performance of the HubLink retriever in scholarly literature search tasks. The framework provides a controlled environment for systematic testing and benchmarking of various KGQA approaches, all embedded within a graph-based RAG-inspired pipeline. This setup enables configurable evaluations and supports the construction of (domain-specific) QA systems. The framework is designed with extensibility in mind, promoting reuse and adaptability within the research community. Following, we provide an overview of the main components of the retrieval framework:

Configuration Component The configuration component manages experiment setup through easily modifiable, `JSON`-based configuration files. It defines and serializes parameters for all system modules, enabling reproducibility and systematic variations in the experiment setup. During initialization, the component parses and validates these configurations, ensuring consistent control over module selection and parameterization across the retrieval framework.

Data Component The data component handles loading, preprocessing, and distribution of metadata and QA pairs from standard `JSON` and `CSV` formats. It serves as the foundation for knowledge base construction and evaluation datasets. Within the framework, it supports structured LLM-based extraction from publication texts, managing both input and output data for downstream retrieval and classification tasks.

Pipeline Component The pipeline component orchestrates the core logic of the question-answering process within the retrieval framework. It implements a modular and customizable graph-based RAG-inspired pipeline consisting of four sequential stages: Pre-Retrieval, Retrieval, Post-Retrieval, and Generation. Each stage is designed to be interchangeable and configurable, enabling flexible experimentation with different models and algorithms. The pipeline is built using the LangChain library and processes input questions, optionally with a topic entity. This design supports semantic enrichment, context-aware retrieval, and answer generation for scholarly search tasks.

Retriever Component The retriever component is responsible for identifying contextually relevant information from a designated knowledge base, such as a knowledge graph or document index. Integrated with the pipeline, it supports two retrieval modes: *document-based retrieval*, which operates on publication texts, and *knowledge graph-based retrieval*, which queries structured data. This modular design allows flexible configuration and integration of different retrieval strategies to support question answering within the framework.

Language Model Component The language model component provides a unified interface for accessing both local and proprietary API-based LLMs. It manages communication between the retrieval framework and the underlying models, handling request execution and response parsing to support downstream tasks such as prompting and answer generation.

Experiment Component The experiment component enables systematic evaluation of retrieval performance within the framework. It automates the execution of QA pipelines based on configurable parameters, iterates over test questions, and computes relevant RAG metrics such as retrieval recall and answer relevancy. Results are stored for reproducibility, and visualized through diagrams and tables to support data analysis and performance benchmarking.

App Component The retrieval framework incorporates a Command Line Interface (CLI) designed to facilitate configuration management, data ingestion, experiment execution, and interactive question answering. This component streamlines the experimental workflow by enabling users (researchers) to interact with the system via terminal commands.

13.4. Evaluation

In this section, we address the validation questions VQ II-3.1 and VQ II-3.2 regarding performance benchmarking. Therefore, we evaluate HubLink with the extrinsic eval-

uation strategy of our evaluation approach (cf. Section 5.3) to systematically assess (1) retrieval quality and (2) answer generation quality. We first describe our evaluation setup (Section 13.4.1), then present the results of both validation questions in (Section 13.4.2).

13.4.1. Evaluation Settings

In this section, we describe the experimental setup for evaluating HubLink against state-of-the-art baselines. First, we introduce the selected baselines (Section 13.4.1.1) for benchmarking. Second, we present the parameter selection and configuration for each retrieval approach (Section 13.4.1.2) using the OFAT method (see Section 2.8.7). For the evaluation, we consider and derive 4 HubLink variants (Section 13.4.1.3). We use the ORKG as infrastructure and knowledge base with 4 KGs variants based on two orthogonal characteristics: path length (i.e., affecting the number of hops) and contribution granularity (i.e., affecting the degree of branching), as described in Section 11.3. This setup enables us to assess the impact of knowledge representation variants of a classification and data schema on retrieval performance. For the evaluation, we also need a dataset with QA pairs and gold standard KG triples. Therefore, we used two datasets for the parameter selection (44 QA pairs) and the final evaluation (170 QA pairs), as described in Section 10.4.3. We constructed the datasets according to our taxonomy (Section 10.3) to derive a balanced dataset across varying degrees of retrieval and question complexity in the queries and thereby avoid bias in the obtained results.

13.4.1.1. Baselines

For the evaluation against state-of-the-art approaches (see Section 13.4.2), we defined specific criteria for baseline selection. Given our research objective on training-free scholarly KGQA, we required baseline approaches to be:

1. LLM-based / embedding-based KGQA systems that accept a question in natural language as input,
2. training-free to ensure fair comparison with HubLink, and
3. accompanied by available and functional source code or a detailed specification of the approach to enable reproducible evaluation.

Our baseline selection process mainly followed a review of two KGQA surveys by Pan et al. [213] and Peng et al. [216]. We identified three approaches that met all our criteria above and represent different paradigms of training-free KGQA: Direct Fact Retrieval (DiFaR) [17] (embedding-based), MindMap [302] (subgraph construction), and FiDeLiS [274] (stepwise reasoning).

Notably, all selected state-of-the-art baseline models have been published in top-tier ACL[1] venues (Association for Computational Linguistics) between 2023 and 2025, thereby ensuring both methodological rigor and validation by the research community.

[1] `https://dblp.org/db/conf/acl/index.html` [Last accessed on 2025-10-15]

Direct Fact Retrieval (DiFaR) [17] indexes KG triples as embeddings. At query time, the approach uses ANN search to retrieve the closest and most relevant triples based on embedding similarity for each query. Retrieved triples serve as context for answer generation, with an optional LLM-based reranking refinement step.

MindMap [302] embeds all KG entities during preprocessing and uses LLM-driven entity extraction with ANN search to identify relevant entities from input questions. It constructs two evidence subgraphs, i.e., shortest paths between entities and 1-hop neighborhoods, then converts them to natural-language descriptions using an LLM to serve as context for answer generation.

FiDeLiS [274] performs beam search from identified graph entry points (topic entities) using LLM-generated strategic plans that extract keywords and convert queries into declarative format. It iteratively retrieves and embeds relational paths, scoring them against keyword embeddings to maintain top-N candidates until the termination criteria or maximum path length are reached, then generates final answers from candidate paths.

13.4.1.2. Parameter Selection and Configuration

HubLink, as well as the selected baseline approaches (Section 13.4.1.1), all require certain parameters to be selected. Due to practical constraints, including both high costs and time requirements for LLM experiments (latency), a fully fledged parameter optimization process is infeasible. Thus, we employ a parameter selection process using the OFAT method [197] (see Section 2.8.7) to optimize retrieval performance based on Recall and Hits@10 metrics. The following list enumerates the individual parameters. Italic parameters served as the baseline for the OFAT method. In the evaluation, we use the parameters that have shown the most promising results using OFAT, highlighted by underlining. For parameter optimization, we utilized a reduced dataset of 44 QA pairs (see Section 10.4.3), following the same construction method based on our question retrieval taxonomy (see Section 10.3) as the full dataset, evaluated against graph variant GV1.

General parameters for all approaches are:

- LLM Closed: *gpt-4o-mini* [209], gpt-4o [208], gpt-o3-mini [205];
- LLM Open: *Qwen2.5-14B* [282], Llama3.1-8B [84]),
- Embedding Models: *mxbai-embed-large* [174], text-embedding-3-large [210], granite-embedding [83]
- Question Augmentation (boolean): (*false*/true)
- Reranking (boolean): (*false*/true).

HubLink-specific parameters are: traversal strategy (*false*/true), extract question components (*false*/true), top paths to keep (*10*/20/30), number of hubs (*10*/20/30), filter output context (false/*true*), diversity ranking penalty (0/0.01/*0.05*/0.1), and path weight alpha (0/3/*5*/9).

Baseline parameters are:

- DiFaR uses number of results (*30*/60/90/120/150) and distance metric (*cosine*/IP/L2);
- MindMap considers final paths to keep (*10*/20/30), shortest paths to keep (*10*/20/30), and neighbors to keep (*10*/20/30);
- FiDeLiS employs top-k (*10*/20/30), top-n (*10*/20/30), and alpha (0.1/*0.3*/0.6).

Besides the above parameters, we derive different variants of **HubLink Configurations** (see Section 13.4.1.3).

13.4.1.3. HubLink Variants and Configurations

For our evaluation regarding the retrieval quality and to address our first evaluation goal (see Section 13.4.2.1), we consider four HubLink variants with the following configurations:

HubLink (T) is the primary configuration, selected through the parameter selection process and expected to achieve the best performance. It uses the graph traversal strategy (Section 13.2.2.3) of HubLink.

HubLink (D) shares the same configuration as HubLink variant (T) but instead uses the direct retrieval strategy of HubLink (Section 13.2.2.2). This allows us to compare the *direct retrieval strategy* with the *graph traversal strategy* using the same parameter configuration.

HubLink (F) is specifically configured for reduced runtime and, thus, provides an economical variant optimized for rapid execution, a crucial factor for practical use. This configuration employs the direct retrieval strategy (HubLink (D)) and limits the number of hubs to 10 per question.

HubLink (O) shares the same configuration as HubLink variants (T) and (D) but is dedicated to open-source / open-weights models, by utilizing the *mxbai-embed-large embedding model* [174] and the *Qwen2.5-14B LLM* [282] to appreciate Open Science principles.

These variants are introduced and used for the evaluation to systematically assess the robustness, efficiency, and adaptability of HubLink across different retrieval strategies and parameter configurations.

13.4.2. Evaluation Results

In this section, we present the empirical results of our evaluation across two main evaluation goals: (1) **retrieval quality** (Section 13.4.2.1) and (2) **answer generation quality** (Section 13.4.2.2). All metrics have been macro-averaged.

13.4.2.1. Evaluating Retrieval Quality

First, we evaluate the **retrieval quality (Evaluation Goal 1)** by answering VQ II-3.1, which we further divide into five questions (VQ II-3.1.1 to VQ II-3.1.5). Our evaluation begins by comparing HubLink's retrieval performance to baseline approaches, focusing on the accuracy and relevance of retrieved contexts; mainly based on the metrics Recall and MAP@10. The influence of different question complexities on performance is then examined, followed by an assessment of HubLink's applicability across various scholarly literature search scenarios (use cases). The impact of typed information embedded in input questions is evaluated, along with HubLink's robustness to structural and lexical variations in the underlying knowledge graph.

To address these questions associated with evaluation goal 1, we use common performance retrieval metrics, as described in Section 2.8.6. For the evaluation, we use the four HubLink variants (Section 13.4.1.3) and the four different knowledge base variants (Section 11.3), all capturing our classification and data schema for SWA research based on the labeled elements of the corresponding gold standard dataset (Chapter 7).

VQ II-3.1.1: What is the overall retrieval performance of the different HubLink variants in comparison to the baselines?

Table 13.1 presents the retrieval performance of HubLink variants and baselines on graph variant GV1. HubLink (T) with the traversal strategy achieves the highest overall performance, outperforming the baselines across all metrics. It also surpasses HubLink (D), which trades topic entity requirements for reduced effectiveness in complex graphs. Notably, HubLink (T) reaches a Recall of 75.4%, more than double that of the next best baseline DiFaR (35.2%), and demonstrates superior ranking quality with a Hits@10 of 51.2% and an MRR@10 of 50.2%. However, its Precision remains modest (24.6%), indicating the presence of irrelevant triples, which may still contribute to answer generation. HubLink (D), using the direct retrieval strategy, shows reduced performance across all metrics. HubLink (F), optimized for runtime with a limited scope, presents a trade-off (w.r.t. the number of Hubs): it achieves the highest Precision (27.8%) and F_1 score (34.4%), but lower Recall and Hits@10. This suggests that retrieving fewer triples can improve precision by increasing the proportion of relevant results. HubLink (O), the open-source variant, generally underperforms, recording the lowest scores across most metrics. An exception is MRR@10 (52.6%), where it ranks correct triples highly when retrieved, despite a lower overall retrieval rate. A comparative analysis reveals that embedding-based methods (HubLink, DiFaR) considerably outperform subgraph-based (MindMap) and stepwise-reasoning (FiDeLiS) approaches in retrieving relevant triples. These findings highlight the effectiveness of dense vector retrieval in large-scale KGQA tasks, despite filtering/ranking limitations.

VQ II-3.1.2: How do different question complexities influence the overall performance?

Table 13.2 and Table A.6 present the performance results for the four HubLink variants and the baselines across the eight question-complexity categories, as defined in our KGQA retrieval question taxonomy (see Section 10.3). The results are based on graph variant GV1. All HubLink variants considerably outperform all baselines for each question complexity

Table 13.1.: Evaluation results of the retrieval performance of HubLink variants and baseline approaches. The results are based on graph variant **GV1**

Approach	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	**0.754**	0.246	0.328	**0.512**	**0.299**	0.502	**0.298**
HubLink (D)	0.709	0.221	0.277	0.436	0.259	0.486	0.273
HubLink (F)	0.649	**0.278**	**0.344**	0.451	0.267	0.473	0.290
HubLink (O)	0.559	0.144	0.188	0.408	0.272	**0.526**	0.222
DiFaR	0.352	0.011	0.022	0.208	0.151	0.297	0.104
MindMap	0.119	0.030	0.045	0.015	0.002	0.013	0.007
FiDeLiS	0.093	0.053	0.063	0.093	0.063	0.103	0.053

category. For *Basic* operations, which involve retrieving a single triple, all HubLink variants achieve their highest Recall and Hits@10 scores, indicating strong performance on simple fact-based queries. HubLink (T) and HubLink (O) also record the highest F_1 scores for these operations. In contrast, HubLink (D) and HubLink (F) perform best on *Comparative* and *Relationship* questions, suggesting variant-specific strengths. Across all operations, Precision and F_1 scores are consistently lower than Recall, reflecting challenges in precisely identifying relevant triples, particularly for *Negation* and *Superlative* operations, which show the lowest scores across all metrics. No consistent trend is observed in MAP@10, MRR@10, or EM@10 across all variants, though *Negation* and *Superlative* operations tend to yield lower values. These findings highlight the need for improved handling of complex logical constructs in KGQA retrieval.

VQ II-3.1.3 How do different use cases, defined by input condition types and expected answer types, impact retrieval performance?

Table A.7 and Table A.8 present the performance results of the different HubLink variants and the baselines across the different use cases (see Section 10.3). The use cases are based on descriptive metadata, each characterized by distinct input condition types (conventional metadata, knowledge-based metadata, or both) and expected answer types (conventional metadata or knowledge-based metadata), forming the competency questions and gold standard answers. Use cases involving conventional metadata input conditions consistently yield higher Recall scores. All HubLink variants consistently outperform the baseline approaches across all use cases. For HubLink, in particular, *Use Case 1* (conventional metadata input and output) and *Use Case 2* (conventional metadata input, knowledge-based metadata output) achieve the strongest Recall across all variants. *Use Cases 5* and *Use Case 6*, which combine mixed conventional and knowledge-based metadata inputs, follow closely. In contrast, queries based solely on knowledge-based metadata (*Use Cases 3* and *Use Case 4*) result in lower Recall, indicating greater difficulty in retrieving relevant triples from purely knowledge-based metadata conditions. Precision and F_1 scores are generally lower for use cases requiring knowledge-based metadata answers, indicating challenges in accurately retrieving relevant content. Ranking metrics (MAP@10, MRR@10, EM@10) show the strongest performance for *Use Case 1*, with no consistent trends across

Table 13.2.: Evaluation results of the retrieval performance for HubLink (T) and the baselines, considering the impact of the different question complexity types. The results are based on graph variant **GV1**.

	Question Complexity	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	basic	**0.917**	**0.382**	**0.480**	**0.917**	**0.445**	0.490	**0.389**
	aggregation	0.810	0.209	0.285	0.497	0.225	0.347	0.240
	counting	0.840	0.275	0.372	0.644	0.357	0.526	0.340
	ranking	0.817	0.321	0.414	0.561	0.360	**0.576**	0.363
	comparative	0.742	0.262	0.366	0.456	0.320	0.560	0.296
	relationship	0.628	0.254	0.314	0.410	0.298	0.528	0.331
	negation	0.584	0.072	0.122	0.244	0.125	0.419	0.144
	superlative	0.656	0.129	0.193	0.319	0.207	0.540	0.237
DiFaR	basic	0.528	0.006	0.006	0.167	0.083	0.083	0.017
	aggregation	0.365	0.008	0.017	0.236	0.199	0.302	0.083
	counting	0.350	0.007	0.017	0.257	0.215	0.267	0.092
	ranking	0.164	0.005	0.013	0.111	0.089	0.190	0.058
	comparative	0.363	0.014	0.029	0.247	0.130	0.331	0.154
	relationship	0.380	0.017	0.032	0.270	0.239	0.545	0.196
	negation	0.364	0.013	0.031	0.219	0.107	0.308	0.131
	superlative	0.346	0.018	0.034	0.113	0.087	0.278	0.075
MindMap	basic	0.278	0.020	0.037	0.056	0.006	0.006	0.006
	aggregation	0.115	0.031	0.046	0.004	0.000	0.004	0.004
	counting	0.182	0.041	0.065	0.004	0.000	0.004	0.004
	ranking	0.105	0.029	0.043	0.016	0.002	0.009	0.008
	comparative	0.026	0.010	0.014	0.005	0.005	0.042	0.004
	relationship	0.116	0.055	0.073	0.018	0.002	0.015	0.013
	negation	0.054	0.020	0.029	0.023	0.003	0.023	0.019
	superlative	0.079	0.030	0.041	0.000	0.000	0.000	0.000
FiDeLiS	basic	0.333	0.208	0.245	0.333	0.306	0.306	0.220
	aggregation	0.035	0.016	0.022	0.035	0.013	0.028	0.017
	counting	0.021	0.012	0.015	0.021	0.010	0.021	0.012
	ranking	0.102	0.037	0.051	0.101	0.038	0.072	0.029
	comparative	0.157	0.067	0.087	0.157	0.099	0.173	0.079
	relationship	0.052	0.044	0.048	0.053	0.033	0.121	0.046
	negation	0.011	0.011	0.011	0.010	0.010	0.062	0.006
	superlative	0.045	0.047	0.045	0.045	0.019	0.062	0.031

Table 13.3.: Evaluation results of the retrieval performance, considering the impact of semi-typed and untyped questions. The results are based on **GV1**.

Semi-Typed	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)							
True	**0.763**	**0.258**	**0.352**	**0.567**	**0.328**	**0.512**	**0.323**
False	0.747	0.233	0.302	0.456	0.268	0.488	0.273
DiFaR							
True	0.256	0.189	0.349	0.011	0.359	0.023	0.128
False	0.155	0.109	0.238	0.010	0.343	0.021	0.078
MindMap							
True	0.006	0.001	0.006	0.029	0.105	0.042	0.005
False	0.024	0.004	0.021	0.031	0.133	0.047	0.010
FiDeLiS							
True	0.108	0.076	0.121	0.065	0.108	0.076	0.062
False	0.075	0.048	0.083	0.039	0.075	0.049	0.043

other cases. Overall, HubLink performs best in conventional metadata-driven scenarios, reflecting current scholarly search practices. However, its strong Recall for knowledge-based queries suggests potential for enabling more content-focused research workflows. Nonetheless, limitations in ranking and filtering for complex queries highlight the need for enhanced post-retrieval mechanisms, such as reranking and contextual refinement, to improve performance in complex query scenarios.

VQ II-3.1.4 How do semi-typed and untyped questions affect the retrieval performance?

Table 13.3 and Table A.9 present the performance results of the different HubLink variants and the baselines considering the impact of semi-typed and untyped questions (examples in Section 10.3). Type annotations help disambiguate entity roles and reduce the candidate search space by filtering irrelevant triples. All HubLink variants consistently outperform the baselines on both question types. The HubLink variants (T), (D), and (F) show improved performance across all metrics when type information is present, with HubLink (F) exhibiting the most notable gains in Precision, F_1, MRR@10, and EM@10. In contrast, HubLink (O), which utilizes open-source/open-weight LLM and embedding model, shows a consistent decline in performance with typed question input, while performing reliably well on untyped questions. These results suggest that the benefit of type information depends on the underlying model and its ability to leverage semantic cues.

VQ II-3.1-5 How robust is the best-performing HubLink configuration to structural and lexical variability across different graph variants in comparison to the baseline?

Table 13.4 presents the retrieval performance for HubLink (T) and the baseline approaches across the four graph variants (see Section 11.3). Table A.10 shows the retrieval

Table 13.4.: Evaluation results of the retrieval performance for the different graph variants

	Graph	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	GV1	0.755	0.246	0.327	0.513	0.298	0.500	0.299
	GV2	0.759	0.276	0.352	0.518	0.333	0.522	0.324
	GV3	**0.812**	0.350	0.423	0.596	0.408	0.650	0.406
	GV4	0.804	**0.393**	**0.452**	**0.597**	**0.425**	**0.661**	**0.444**
DiFaR	GV1	0.352	0.011	0.022	0.207	0.150	0.295	0.104
	GV2	0.314	0.009	0.019	0.199	0.142	0.268	0.096
	GV3	0.523	**0.017**	**0.035**	0.302	**0.230**	0.442	0.154
	GV4	**0.528**	**0.017**	**0.035**	**0.304**	0.228	**0.449**	**0.158**
MindMap	GV1	0.119	0.030	0.045	0.015	0.002	0.013	0.007
	GV2	0.093	0.025	0.037	0.008	0.001	0.006	0.005
	GV3	0.133	0.043	0.061	**0.030**	0.007	0.023	0.015
	GV4	**0.127**	**0.044**	**0.062**	**0.030**	**0.010**	**0.036**	**0.018**
FiDeLiS	GV1	0.092	0.052	0.063	0.092	0.062	0.103	0.053
	GV2	0.099	0.055	0.064	0.099	0.065	0.110	0.054
	GV3	0.259	0.114	0.139	0.259	0.150	0.240	0.112
	GV4	**0.276**	**0.121**	**0.142**	**0.276**	**0.156**	**0.248**	**0.117**

performance of HubLink (T) and the baselines, considering the impact of the different number of hop counts based on graph variant GV1. HubLink (T) demonstrates superior robustness to graph variations compared to the baselines. Shorter paths consistently improve performance across all approaches, especially for Precision and the ranking metrics. While all approaches benefit from shorter paths, baseline approaches show substantial degradation with longer paths, underscoring their limitations in multi-hop reasoning. In contrast, HubLink maintains stable and superior performance regardless of path length. Despite a slight decrease in precision with longer paths, HubLink (T) remains the most effective at retrieving deeply embedded information. Thus, the schema-agnostic design of HubLink allows seamless application across all graph variants without configuration changes, confirming its adaptability to different entity types and structures.

> ***Summary VQ II-3.1*** *(Retrieval Quality)*
> *HubLink considerably improves retrieval performance in scholarly literature search tasks compared to baseline KGQA approaches, particularly in terms of Recall and Precision. It more than doubles the Recall of the next-best baseline, indicating stronger retrieval of relevant triples. HubLink shows superior robustness across graph structures and hop counts, outperforming baselines in multi-hop reasoning and adapting effectively to complex graph configurations. However, ranking precision remains a limitation, especially in complex queries, where irrelevant information is more frequently retrieved. Minor effects from question type annotations were observed, with limited influence on overall retrieval quality.*

Table 13.5.: Evaluation results of the answer generation quality. The results are based on graph variant **GV1**.

Approach	FC-Reca.	FC-Prec.	FC-F_1	BERT-Reca.	BERT-Prec.	BERT-F_1
HubLink (T)	**0.543**	**0.301**	**0.361**	0.678	0.515	0.580
DiFaR	0.387	0.290	0.321	**0.702**	0.588	**0.635**
MindMap	0.203	0.212	0.184	0.652	0.625	0.633
FiDeLiS	0.131	0.201	0.154	0.516	**0.629**	0.562
Approach	ROG-Reca.	ROG-Prec.	ROG-F_1	Str. Sim.	Sem. Sim.	BLEU
HubLink (T)	**0.757**	0.298	0.373	0.261	0.761	0.105
DiFaR	0.674	0.374	**0.448**	**0.338**	**0.772**	**0.160**
MindMap	0.487	0.432	0.397	0.296	0.682	0.105
FiDeLiS	0.195	**0.503**	0.251	0.202	0.499	0.046

Approach	Answer Relevancy	Instruction Following	Faithfulness
HubLink (T)	**0.570**	**0.653**	0.445
DiFaR	0.203	0.312	**0.645**
MindMap	0.545	0.388	0.396
FiDeLiS	0.432	0.388	0.112

13.4.2.2. Evaluating Answer Generation Quality

Second, we evaluate the answer generation quality (**Evaluation Goal 2**) by answering VQ II-3.2, which we further divide into four questions (VQ II-3.2.1 to VQ II-3.2.4). To address these questions associated with evaluation goal 2, we use common performance answer generation metrics (both traditional and LLM-as-a-judge metrics), as described in Section 2.8.6. For the evaluation, we use the best HubLink variant (T) (based on the parameter selection and retrieval quality) against the baselines on graph variant GV1. Table 13.5 presents the evaluation results using the metrics Factual Correctness (FC), BERTScore (BERT), ROUGE (ROG), String Similarity (Str. Sim.), Semantic Similarity (Sem. Sim.), BLEU, Answer Relevancy, Instruction Following, and Faithfulness.

VQ II 3.2.1: How semantically and factually consistent are the generated answers based on the retrieved triples of HubLink in comparison to the baselines?

Factual correctness and semantic consistency of answers based on retrieved triples represent a fundamental evaluation aspect for assessing the answer generation quality. This study assesses the factual correctness and semantic consistency of answers generated by HubLink compared to baseline KGQA approaches. While HubLink achieves the highest factual correctness recall (54.3%), it falls below its retrieval recall (75.4%), indicating substantial fact loss during generation. Its high ROUGE-1 recall (75.7%) suggests lexical overlap rather than accurate fact integration. In contrast, DiFaR and MindMap demonstrate better alignment between retrieved and generated content, reflecting more efficient fact retention. HubLink's BERTScore recall (67.8%) is behind DiFaR (70.2%), and lower precision across metrics highlights structural and lexical divergence from reference answers. Although

HubLink shows slightly higher factual precision (30.1%) than its retrieval precision (24.6%), it still lags behind DiFaR (29.0%), suggesting the inclusion of extraneous content. Overall, HubLink produces comprehensive but verbose answers, which reduces factual focus and consistency, underscoring the need for improved fact integration and answer conciseness.

VQ II 3.2.2: How relevant to the questions are HubLink's answers in comparison to the baselines?

Answer Relevancy is a crucial evaluation aspect to assess whether the response is relevant to the posed question. HubLink achieves the highest score (57.0%), outperforming MindMap (54.5%), FiDeLiS (43.2%), and DiFaR (20.3%). Importantly, the Answer Relevancy metric assesses semantic fit rather than factual accuracy, meaning relevant answers may still contain hallucinations or unsupported claims. While HubLink demonstrates strong relative performance, 43% of its answers lack optimal semantic alignment between questions and generated answers, indicating room for improvement.

VQ II 3.2.3: How well does HubLink follow instructions in the question in comparison to the baselines?

Instruction Following assesses how well an approach follows explicit answer-formatting instructions (e.g., ordering or aggregation) embedded in (complex) scholarly questions. Our evaluation using the Instruction Following metric shows that HubLink achieves the highest score (65.3%), outperforming MindMap / FiDeLiS (38.8%) and DiFaR (31.2%). Despite its relative advantage, HubLink fails to adhere to instructions in approximately one-third of the cases. This indicates that while our retrieval approach HubLink demonstrates superior instruction compliance compared to the baselines, further refinement is needed to ensure consistent and reliable adherence.

VQ II 3.2.4: How well are the answers of HubLink aligned with the retrieved contexts in comparison to the baselines?

Faithful answer generation requires strict grounding in the retrieved context, avoiding unsupported content. Our evaluation using the Faithfulness metric shows that DiFaR achieves the highest score (64.5%), indicating strong contextual adherence. In contrast, HubLink (44.5%), MindMap (39.6%), and FiDeLiS (11.2%) demonstrate progressively weaker grounding in the retrieved context. These results suggest that while HubLink performs moderately well, it still exhibits limitations in constraining its outputs to the provided evidence, often introducing unsupported or structurally divergent content.

> ***Summary VQ II-3.2*** *(Answer Generation Quality)*
> *HubLink demonstrates strong semantic alignment with user queries, outperforming baselines in Answer relevancy and Instruction Following. However, it exhibits limitations in fact integration and semantic consistency, often including irrelevant information and deviating structurally from reference answers. Compared to DiFaR, HubLink shows weaker contextual grounding. Key areas for improvement include enhancing faithfulness and reducing unsupported claims.*

13.5. Threats to Validity

In this section, we discuss potential threats to validity (TtV) of our experimental design and setup by referencing the TtV classes defined in our classification and data schema (see Table 7.1), and outline corresponding mitigation strategies.

External Validity refers to the generalizability of experimental results beyond the specific study context. In this regard, we can identify two main primary threats concerning the evaluation. The KGQA dataset may not fully reflect the diverse information needs of researchers for knowledge access. To mitigate this, we systematically generated (competency) questions using a structured taxonomy covering multiple query types and six representative use cases (see Section 10.3), ensuring broad coverage of realistic scholarly user scenarios. Another threat to external validity concerns the use of experimental settings or tools that do not accurately reflect actual real-world usage conditions. To address this, we developed a retrieval framework based on the state-of-the-art RAG paradigm and applied standard evaluation metrics (see Section 2.8.6) for assessing the retrieval and answer generation quality with established formulas and implementations to ensure methodological rigor and comparability with related work and related approaches.

Internal Validity assesses whether observed effects in an experiment are caused by the treatment and not by confounding factors. In our study, potential threats include inconsistencies in the adaptation of baseline retriever implementations and potential instrumentation issues. To mitigate these risks, we developed a custom retrieval framework (see Section 13.3) for executing the experiments and integrating baseline retrievers based on their original descriptions and source codes with minimal adaptations.

Construct Validity refers to the extent to which the experimental constructs are well-defined and adequately operationalized. In our study, this risk is considered low, as the evaluation design is informed by recent surveys on RAG-based evaluations and is grounded in the RAGAs framework. Nevertheless, certain threats and limitations remain. First, the experiments include only a subset of possible retriever configurations, which does not fully represent the performance space. This constraint was necessary to ensure a manageable and practical scope and time resources. Second, the reliance on automated metrics without human evaluation introduces a potential measurement bias in terms of mono-method bias. We mitigated this risk by using a variety of traditional performance metrics for RAG as well as LLM-as-a-judge (see Section 2.8.6). Third, the use of the OFAT method may overlook interactions between parameters. However, this method was chosen for its interpretability supported by systematic parameter selection. Overall, the design and methodological choices support the validity of the constructs and their operationalization.

Confirmability and Repeatability are combined discussed as potential threats. Confirmability refers to the extent to which experimental findings are free from researcher bias and grounded in the data, while repeatability concerns the ability to reproduce results under similar conditions. In our study, potential biases may arise from the selection of the

baseline systems, interpretation of ambiguous results, or the design of evaluation scenarios that could accidentally favor our approach. To mitigate these risks, we used experiment tracking tools and customized visualizations. All experimental artifacts, including raw results, processing scripts, and visualization tools, are provided in a replication package. We report both favorable and unfavorable research results. The state-of-the-art baselines were selected through a review of recent KGQA literature (see Section 13.4.1.1) to ensure relevance and fairness. A primary threat stems from the inherent non-determinism of LLMs, which may yield varying outputs across runs. While absolute values may fluctuate, we expect consistent performance trends across the retrieval approaches. To support reproducibility, the retrieval framework captures and preserves complete experimental configurations, enabling exact replication of the setup and reducing variability to the stochastic nature of LLMs. Together, these measures strengthen the transparency, reliability, and reproducibility of our research findings.

13.6. Conclusion

This chapter addressed key limitations in current retrieval systems (see Section 14.2.4), including schema dependence, reliance on training data, and the lack of standardized and systematic evaluation of retrieval approaches (cf. extrinsic evaluation strategy in Section 5.3). In response, we introduced HubLink, a schema-agnostic and training-free retrieval approach tailored for scholarly KGQA tasks by unifying the pros and cons of LLMs and RKGs (see Section 10.1). By transforming RKGs into semantically coherent hubs, HubLink enables modular reasoning and transparent provenance tracking. The embedding-based retrieval uses capabilities to adapt to evolving graph structures without requiring predefined and fixed schemas. This is in alignment with the objective of Part II that deals with evolving classification and data schemas in (hierarchical) knowledge representations. The evaluation on the ORKG using SWA metadata (i.e., descriptive metadata in terms of bibliographic and knowledge-based data) based on the labeled dataset (Chapter 7) demonstrated that HubLink consistently outperforms state-of-the-art approaches, namely DiFaR, MindMap, and FiDeLiS (all published in top-tier ACL venues), across multiple metrics, particularly in handling complex queries (i.e., user questions). The evaluation, grounded in a taxonomy-based QA dataset (see Section 10.3), confirmed HubLink's robustness and effectiveness across six scholarly use cases (metadata types differently combined). HubLink represents a major advancement in scholarly information retrieval by enabling direct and precise access to structured knowledge, embedded within `RDF`-based RKGs. Its independence from schema constraints and training data makes it highly adaptable to dynamic research environments and research fields beyond SE. Moreover, the approach's ability to trace the source (persistent identifier of research artifact) of retrieved information supports the transparency and reliability required in scientific research contexts. While the results confirm HubLink's superior performance, they also highlight areas for future improvement, including refining context relevance ranking and enhancing response generation. Overall, this work contributes a novel and practical solution to scholarly KGQA, which is embedded in a graph-based RAG-inspired pipeline, paving the way for more efficient knowledge access.

Part IV.

Reflections

14. Related Work

In this chapter, we provide an overview of the related work to this doctoral thesis. To this end, we discuss related approaches and research gaps according to the main contributions and the objects under study in Part II (knowledge description and representation) and Part III (knowledge persistence and access).

14.1. Knowledge Description and Representation

In the following sections, we review related work that is aligned with the individual contributions presented in the first part of this doctoral thesis. Therefore, each subsection is explicitly aligned with a specific contribution, allowing for a focused and coherent discussion. For consistency and ease of comparison, the structure of this main section follows the same order as the presentation of our contributions (C1–C4 in Part II).

14.1.1. Evaluation Approaches and Metrics for Classification Schemas

Our first contribution constitutes an evaluation approach for (hierarchical) classification and data schemas. This approach is a (sequential) multi-step evaluation process using the GQM approach (cf. Chapter 5) in an incremental way to support maintenance, evolution, benchmarking, and reproducible classification design. In contrast to state-of-the-art approaches, we aim to operationalize the research design to guide researchers through such an evaluation. In the following, we present contemporary approaches for creating and evaluating classification schemas in the research domains of Software Engineering, Information Systems, Information Science, and KGs evaluation. Further, we identify limitations and drawbacks and argue how to overcome them in our approach.

Evaluation of Classification Schemas in Software Engineering Research In the context of SE research, we identified two primary approaches relevant to our research objectives.

First, Ralph [231] provides structured and methodological principles for developing and evaluating theories and taxonomies (also known as classifications/classification schemas) as essential artifacts for organizing domain knowledge and enabling theory development in SE. These guidelines encompass a multi-step procedure (cf. Figure 14.1, (a)), including recommendations on appropriate research methods for data collection and analysis. The author identifies three core properties (quality criteria) that any good and robust taxonomy should exhibit: (1) The class structure captures both the similarities and dissimilarities among instances; (2) class membership allows for the inference of numerous relevant properties of an instance beyond those required for classification; and (3) the classification effectively serves the specific purpose for which it was designed.

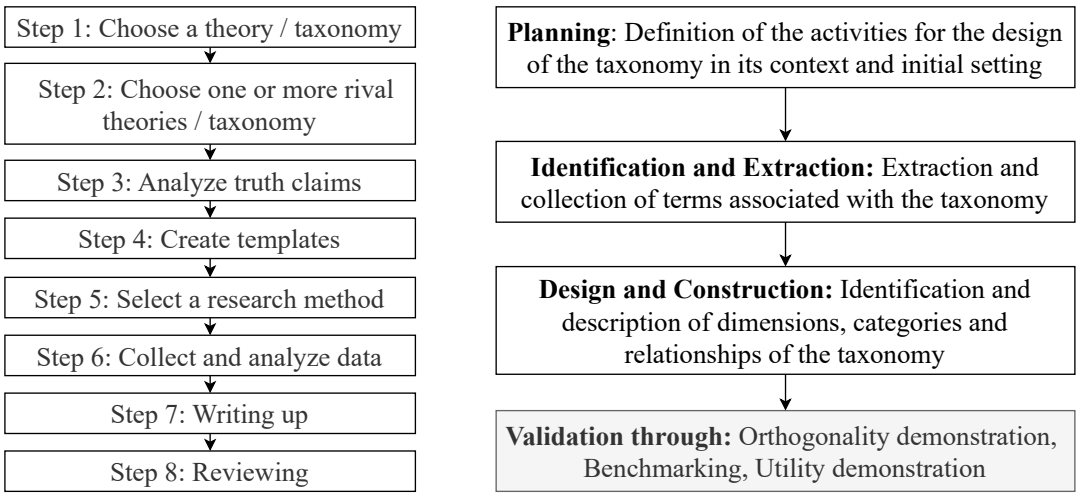

Figure 14.1.: Development and evaluation process for taxonomies in Software Engineering according to (a) Ralph [231] and (b) Usman et al. [292]

Second, Usman et al. [292] define four phases in the revised method for taxonomy development, building upon the foundational work by Bayona-Oré et al. [21], as depicted in Figure 14.1 (b). Each phase encompasses specific activities intended to guide the construction and evaluation of taxonomies within the context of SE. While the method includes a dedicated validation phase to strengthen the reliability and usefulness of taxonomies, its scope encompasses three primary strategies: orthogonality demonstration (i.e., assessing the independence of classification dimensions), benchmarking (i.e., comparative analysis against existing taxonomies), and utility demonstration (i.e., applying the taxonomy to classify representative subject matter examples). For these strategies, the authors refer to the works by Smite et al. [270] and Wheaton [303] (technical report, 1968).

In conclusion, both works highlight the importance of classification schemas within the SE research community. Although the authors outline several strategies for validation, they do not provide concrete metrics, operational criteria, or procedural guidelines to support their practical application. This lack of specificity renders the validation phase methodologically weak and insufficiently defined, thereby limiting its reproducibility and reducing the consistency and comparability of evaluations across studies. More generally, the proposed approaches offer only broad recommendations and illustrative steps concerning the processes involved in developing and evaluating classification schemas (also referred to as taxonomies in SE). However, they do not supply detailed methodological guidance for the design of evaluation procedures—such as the formulation of evaluation questions, the specification of assessment criteria, or the selection of appropriate metrics. Instead of providing operationalized frameworks, the approaches remain at a conceptual level, presenting high-level templates and methodological overviews in a compendium-like fashion. As a result, they offer limited practical utility by lacking actionable support

<table>
<tr><td rowspan="3">Who? Subject of evaluation</td><td>Involvement in taxonomy building</td><td colspan="3">Subject has been involved in taxonomy building</td><td colspan="3">Subject has not been involved in taxonomy building</td></tr>
<tr><td>Background</td><td colspan="3">Academics</td><td colspan="3">Practitioners</td></tr>
<tr><td>Experience</td><td colspan="3">Domain</td><td colspan="3">Method</td></tr>
<tr><td rowspan="3">What? Object of evaluation</td><td>Type</td><td colspan="3">Real-world</td><td colspan="3">About real-world</td></tr>
<tr><td>Involvement in taxonomy building</td><td colspan="2">Object has been used in taxonomy building</td><td colspan="2">Object has been used in taxonomy building and evolved since then</td><td colspan="2">Object has not been used in taxonomy building</td></tr>
<tr><td>Coverage</td><td colspan="2">Exhaustive</td><td colspan="2">Selective</td><td colspan="2">Representative</td></tr>
<tr><td rowspan="3">How? Method of evaluation</td><td>Approach</td><td colspan="3">Quantitative</td><td colspan="3">Qualitative</td></tr>
<tr><td rowspan="2">Method</td><td>Logical argument</td><td>Expert interview</td><td>Focus group</td><td>Survey</td><td>Delphi technique</td><td>Sorting</td></tr>
<tr><td>Log diary</td><td>Case study</td><td>Illustrative scenario</td><td>Action research</td><td>Experiment</td><td>...</td></tr>
</table>

Figure 14.2.: Taxonomy evaluation framework (Figure adapted from Szopinski et al. [277])

for researchers who aim to put rigorous and comparable evaluations of SE classification schemas into practice.

Evaluation Approaches in Information Systems Within the research field of Information Systems, Szopinski et al. [277, 278] investigate evaluation techniques for taxonomies. Similar to Ralph [231], Szopinski et al. [277] regard taxonomies as a tool to analyze and understand a domain. In IS, they also observe a lack of guidance for researchers on how to rigorously evaluate taxonomies. In turn, they present a framework focusing on three main questions: how to evaluate, what to evaluate (the object under study), and who evaluates (the subject of evaluation). In a systematic literature review, Szopinski et al. [278] identified 54 papers that report on taxonomy evaluation criteria, i.e., quality criteria of a taxonomy, collecting 43 different evaluation criteria. *Usefulness* emerges as the most frequently applied evaluation criterion (i.e., “the degree to which the artifact positively impacts the task performance of individuals” [278]). Moreover, we can observe that the terms are not used consistently, as we can find synonyms like "mutually [sic] exclusiveness" [278] and "distinctiveness" [278]. Further, Kundisch et al. [165] extend the aforementioned works and build upon Nickerson et al. [202]. They propose an extended taxonomy design process where the authors consolidate the previous works and review evaluation methods. Still, these works lack operational guidance and a structured method design.

Evaluation Approaches in Information Science Likewise, in Information Science, Bedford [24] identifies two evaluation issues with regard to classification schemas: the evaluation of the classification schema itself and the evaluation of how well the schema supports classification decisions. Both require their own framework and context for evaluation. Here, Ranganathan’s principles of classification [234] are employed as a framework for quality attributes, i.e., exclusiveness, uniqueness, relevance, ascertainability, consistency, affinity, decreasing extension, context, currency differentiation, and exhaustiveness. In our approach, we aim to cover all of them while providing operational guidance for researchers. Consequently, while each of the aforementioned research fields considers evaluation criteria, there is no mapping to evaluation goals or metrics for data analysis.

Table 14.1.: Quality assessment dimensions for knowledge graphs according to Hogan et al. [103] (presented in the extended version [102])

Dimension	Sub-Dimension	Definition
Accuracy	Syntactic Accuracy	Conformance of data values to domain grammar and data model.
	Semantic Accuracy	Correctness of values relative to real-world facts / phenomena.
	Timeliness	Degree to which data reflects the current real-world state.
Coverage	Completeness	Presence of all required schema elements, property values, and instances.
	Representativeness	Absence of systemic bias in included data, schema, or reasoning.
Coherency	Consistency	Freedom from logical contradictions under defined entailment.
	Validity	Conformance to explicit constraints (e.g., shape expressions).
Succinctness	Conciseness	Exclusion of irrelevant schema/data elements.
	Representational Conciseness	Avoidance of redundancy in constructs or entities.
	Understandability	Human-readability and clarity of entity representation.

Evaluation of Knowledge Graphs KGs in general, as an additional knowledge representation technique (Section 2.5), are frequently constructed from heterogeneous and imperfect sources. As a result, they commonly exhibit characteristics such as incompleteness, redundancy, and factual errors. Consequently, post-construction quality assessment and evaluation of KGs are critical to establish their fitness for purpose and determine appropriate use cases. According to Hogan et al. [103, 102], this evaluation integrates different main quality dimensions (and their sub-dimensions) and quantitative metrics (partly provided), extending traditional database quality concepts into the KG domain.

The **accuracy** dimension evaluates the correctness of KG content in terms of entities and relations. *Syntactic accuracy* measures compliance with data model and domain constraints. *Semantic accuracy* assesses factual alignment with reality. *Timeliness* indicates the currency of the KG relative to real-world state. The **Coverage** dimension addresses the inclusion of all domain-relevant information through *completeness* (extent of representation by determining schema, property, population, linkability completeness) and *representativeness* (absence of systemic bias). The **Coherency** dimension ensures logical soundness, with *consistency* checking for contradictions and *validity* ensuring conformance to formal constraints such as shape expressions. The **Succinctness** dimension promotes relevance and clarity, encompassing *conciseness* (exclusion of irrelevant content), *representational conciseness* (avoidance of redundancy), and *understandability* (clear, human-readable, multilingual labeling). Table 14.1 provides an overview of these different properties (quality dimensions) and their definitions for KG quality assessment.

While this work provides a valuable list of quality criteria for KG evaluation, it fails to map these criteria to the specific KG functions and use cases, i.e., the concrete purpose, (e.g., retrieval, metadata cataloging, etc.). In contrast to the specification of a classification schema, the evaluation of KGs considers additional aspects like the populated data, e.g., in terms of the sub-dimension *timeliness*, which reflects whether the data is up-to-date.

Properties for Language Evaluation and their Usage Guizzardi's evaluation framework for conceptual modeling languages introduces four key properties, namely *soundness*, *lucidity*, *laconicity*, and *completeness*, to assess the ontological quality of modeling constructs. These

properties ensure that a language accurately reflects real-world phenomena (soundness), presents concepts clearly (lucidity), avoids redundancy (laconicity), and fully captures the relevant domain (completeness) [88]. The work by Ananieva et al. [9] on unification metrics aligns with these properties by addressing the semantic fragmentation in variability modeling in space and time across domains. To this end, their approach seeks to unify terminologies and modeling constructs by identifying semantically equivalent elements and structuring them into a coherent conceptual model. We adapted the properties and metrics in our evaluation approach for hierarchical classification and data schemas as internal evaluation strategy to analyze the structure's suitability (cf. Section 5.3) w.r.t. the properties *generality* and *appropriateness* embedded in the GQM approach.

Metrics for (sustainable) trade-off decisions For developing software, sustainability as a quality attribute encompasses the following five key dimensions: *environmental*, *economic*, *technical*, *social*, and *individual* [22, 23]. Likewise, trade-off decisions are a fundamental aspect of SE, as the discipline inherently involves balancing competing goals, constraints, stakeholder interests, and so on. These decisions are crucial, as software systems are increasingly expected to fulfill a broad range of quality attributes, especially in the sustainability dimensions. For this, we proposed combined metrics to consider performance aspects and energy consumption when applying LLMs for classical NLP tasks in our evaluation approach (cf. Evaluation Step 2 *applicability*). Similar to our approach, Hasan et al. [95] propose the Robustness Carbon Trade-off Index (RCTI), a novel metric designed to quantify the trade-off between model robustness and carbon emissions in machine learning experiments. They demonstrate the utility of RCTI through an evasion attack scenario, analyzing the relationship between adversarial robustness and the associated carbon footprint. Subsequently, the authors extend this metric by incorporating an economic perspective, introducing the Cost per unit of Robustness Change (CRC) [94], thereby capturing a three-way trade-off among robustness, carbon emissions, and economic cost. However, we identify practical limitations and problems due to implicit, partially domain-specific assumptions and the risk of division by 0, making them less applicable.

Method Construction and Multi-step Evaluation Existing approaches for the evaluation of hierarchical classification schemas often lack a systematic methodology that accounts for both the conceptual design and the operational maturity of the evaluation process. A key gap lies in the methodological construction of evaluation approaches. Our method builds upon the GQM paradigm, which provides a structured framework for deriving metrics from high-level goals through intermediate questions (cf. Section 2.8.1). Although GQM is used in SE research and quality assessment, its application to hierarchical classification schema evaluation remains underexplored. In particular, the formulation of template questions that guide the evaluation across different abstraction levels is not commonly addressed in existing literature. Furthermore, our approach introduces a multi-step evaluation strategy that reflects varying degrees of abstraction and maturity. This is inspired by the work of Böhme & Reussner "Validation of Predictions with Measurements" [31], which emphasizes the importance of aligning evaluation techniques with the predictive and empirical maturity of the system under study in a multi-level approach. Another notable

gap is the lack of orthogonal evaluation strategies that correspond to the conceptual, operational, and quantitative levels defined in GQM. While GQM provides a theoretical foundation for such stratification, we found no existing work that explicitly decomposes evaluation strategies along these orthogonal dimensions in a way that supports hierarchical classification schema analysis. In addition to this, we formulate template research questions for each evaluation goal and corresponding properties to ease the formulation of appropriate questions for the method design. In summary, although foundational concepts such as GQM and multi-level validation exist, their integration into a coherent evaluation approach for hierarchical classification schemas is still missing in the literature.

Conclusion In conclusion, we observed a lack of (operational) guidance for researchers (and reviewers) in the state-of-the-art for the concrete method design in evaluating classification schemas that goes beyond template workflows, listing research methods in a compendium style, and evaluation criteria for quality assessment. Our work aims to address these limitations by proposing a structured, template-driven, and maturity-aware evaluation approach to support and foster maintenance and evolution. To address this issue, our approach explicitly maps evaluation goals to corresponding quality criteria, which represent evaluation questions or hypotheses. These criteria are further linked to specific metrics (e.g., trade-off metrics that account for performance and sustainability considerations) following the GQM paradigm. Along the quality criteria and metrics, we further define evaluation strategies to incorporate the method design and structure the complexity of the evaluation of classification schemas. In this way, we enable maintenance and benchmarking of classification concepts over time regarding their *structure's suitability* (internal and external), *applicability* (manual and automated), and their *purpose* (intrinsic and extrinsic).

14.1.2. Guided Tool Support for Literature Studies in Software Engineering

Our second contribution constitutes a structured documentation process, especially to support the evaluation approach (C1). To the best of our knowledge, we are the first to propose a *tool-supported guideline* specifically designed to facilitate the setup and execution of literature studies in SE in the data extraction phase (cf. Table 6.1) that adhere to Open Science principles. Our approach supports both novice and experienced (literature) reviewers in managing and coordinating large-scale review efforts. It uses machine-readable formats to enable computational support and integrates features of modern version control systems to enhance collaboration, traceability, and knowledge sharing.

Our guideline for tool support complements current conceptual guidelines in SE research for the planning and conduction of systematic literature studies, such as Kitchenham and Charters [147] or Petersen et al. [221]. It also extends the functionality of existing resources like the Systematic Review Toolbox[1], originally developed by Christopher Marshall in 2014 and updated in 2022 to reflect advances in evidence synthesis methodology and to support various families of literature studies [127]. Our contribution addresses persistent challenges in the data extraction phase, particularly those related to collaboration. Specifically, the

[1] `https://www.systematicreviewtools.com/` [Last accessed on 2025-08-17]

guideline is designed to mitigate issues such as insufficient support for multi-reviewer workflows, a lack of traceability in decision-making, and difficulties in achieving consensus. By providing structured, tool-supported guidance, we aim to reduce inconsistencies and enhance transparency throughout the review process. Importantly, we move beyond the use of simple spreadsheet formats for data extraction, which are often inadequate for representing complex data structures and are prone to information loss. Instead, our approach supports richer data representations that are better suited for computational analysis and reproducibility.

Positioned at the intersection of conceptual methodology and practical tool support, our approach aligns with experience-based guidelines such as that of Kuhrmann et al. [164], which emphasize pragmatic strategies for study design, especially in relation to data collection and selection procedures. Our work builds on these foundations by offering a scalable and transparent solution for improving the quality and efficiency of the data extraction phase in SE literature studies. With our contribution (C2), we address several meta-research topics (cf. Section 2.1.1). This includes *Methods* by supporting EBSE studies through operational, tool-supported guidance. This approach fosters transparent documentation during the data extraction phase, enabling large-scale reviews and supporting educational purposes, while also contributing to the meta-research topics *Reporting* and *Reproducibility* through systematically deriving replication packages. Our guidelines may also set new standards for supporting reviewers in artifact evaluation tracks for *Evaluation* and set quality criteria for *Incentives* reflected in artifact badging systems.

14.1.3. Classification Schemas and Meta-Research Studies in Software Architecture Research

Our third contribution constitutes a general knowledge description approach to specify papers as research artifacts according to their *findings*, *their validity*, and *evidence* (i.e., their relation to research data and software), thereby following a more holistic and modular view focusing on statements in a paper. We instantiate our approach to SWA research as a domain-specific use case example in SE. While the employed research methods and topics have been investigated before, we identified this gap, comprising considerations about broader evaluation strategies and replication aspects. In particular, we set, among other contributions, a new state-of-the-art standard w.r.t. the *Research Objects*, provide insights about replicability issues, and present concrete proposals for the SWA research community based on an analysis of state-of-practice in published ECSA and ICSA papers from 2017–2021 in the sense of the objective of meta-research and its corresponding translational framework (cf. Hardwicke et al. [93]).

In the following, we identify and present the main areas of related work that contextualize our work and approach. In particular, we review and discuss: (1) secondary and meta-research studies on SWA research, (2) secondary studies on evaluation methods in SE, (3) investigations addressing validity and replication issues within SE research, and (4) guidelines for research practices.

Secondary Studies on Software Architecture Research First, we consider literature studies on state-of-practice in SWA research as highly related to our work.

Qureshi et al. [228] (2013) conducted a systematic literature review to synthesize empirical evidence in SWA research, aiming to identify trends and mature sub-areas of SWA, recurring patterns, and knowledge gaps based on 247 primary research studies. Their study categorizes SWA topics and areas, such as software architecture evaluation, software architecture design, and architecture documentation / description, and analyzes the prevalence of empirical methods based on the types of evidence used. However, their review focuses on shallow metadata and does not explore interdependencies or conceptual relationships among the identified topics, limiting its capacity to reveal structural insights and research knowledge description within the field.

Galster and Weyns [72] (2016) focus on the state of empirical research in SWA by examining the role of empirical methods in validating architectural approaches through the categorization of 667 full technical papers published at SWA venues. They emphasize the importance of empirical studies for understanding architectural phenomena, building confidence in proposed solutions, and establishing the credibility of research outcomes. Their analysis includes the frequency of empirical methods applied, the involvement of human participants, and the treatment of validity concerns in SWA publications.

While both studies contribute valuable perspectives on empirical practices in SWA, our approach extends them by incorporating both empirical and non-empirical evaluation methods. Furthermore, we address replication aspects and trustworthiness concerns while considering traditional validity threats. By doing so, we aim to provide a more comprehensive understanding of methodological rigor and reproducibility in SWA, focusing on the modular description of research findings, their validity and evidence.

Further Meta-Research Studies in Software Architecture Further, we consider other meta-research studies in SWA research beyond applying literature reviews as related.

One of the earliest works in this area is Mary Shaw's paper (2001) "The Coming-of-Age of Software Architecture Research" [260], which critically examines the maturation of SWA as a distinct research discipline within SE. Drawing upon Redwine and Riddle's technology maturation model, Shaw articulates the transition of SWA from an informal, practice-driven activity to a structured research field characterized by formal methods, analytical rigor, and institutional support. In this paper, Shaw also introduced the taxonomy for describing SE research, comprising the dimensions of Research Setting, Research Product, and Technique. These dimensions were later refined and renamed in her highly cited paper "Writing Good Software Engineering Research Papers" [167], where they appear as Type of Question, Type of Result, and Type of Validation (see below). Not only were the labels of the dimensions adapted, but the classification categories themselves were slightly modified to better reflect the evolving nature of SE research.

Galster and Weyns [73] (2018) build upon their earlier work [72] to discuss key challenges in empirical software architecture (SWA) research. These include poorly designed and communicated studies, a lack of replication efforts, and limited practical relevance, as also emphasized in their follow-up publication. Therefore, they consulted 455 program

committee members of all editions of major SWA conferences (i.e., CBSE[2], ECSA[3], ICSA[4], QoSA[5], and WICSA[6]) in a questionnaire to get a deeper understanding of perceptions of SWA researchers, receiving 105 valid responses (i.e., a response rate of ~23%). They look at SWA as a research subfield of SE, noting that these issues can also be observed in the larger context of SE research. In a subsequent study [71], published in 2023, the same authors further explored community perceptions regarding empirical research practices in SWA. Their investigation focused on four key areas: the application of empirical methods, the role of human participants, internal and external validity, and replication practices. Using a mixed-methods approach, they analyzed both qualitative and quantitative data, revealing that consensus on empirical standards remains limited within the SWA community.

Collectively, these studies aim to provide an overview of empirical research in SWA by integrating diverse evaluation strategies and knowledge sources, including both descriptive literature data and community perceptions.

Also, a more recent meta-research study by Kazman et al. [143] (2023) examines replication artifacts and explores the potential for mining and sharing datasets to enhance reproducibility in SWA research. The study underscores the importance of accessible and reusable data resources in promoting empirical rigor and transparency within the field.

Secondary Studies on Evaluation Methods in Software Engineering Research We consider literature studies addressing evaluation methods as related work. The following papers discuss general SE evaluation approaches that are not necessarily specific to SWA research.

Shaw [261] proposed a set of validation types, namely analysis, evaluation, experience, example, and no validation or blatant assertion, to categorize strategies for investigating research problems in SE. To assess the applicability of these categories, a literature study of ICSE 2002 submissions was conducted, encompassing both accepted and rejected papers. This initial validation effort was later extended by Theisen et al. [283], who performed a literature review of ICSE 2016 submissions, and by Bertolino et al. [29], who analyzed accepted papers from ICSE, ESEC/FSE[7], and ASE[8] between 2012 and 2016. These follow-up studies consolidated Shaw's categorization schema by applying it to a broader and more recent body of literature. Unlike our review, which focuses on explicitly named evaluation methods, these studies cluster concrete evaluation techniques (e.g., case study, grounded theory) under broader validation types, describing them based on their methodological characteristics rather than naming them directly.

In a related effort, Glass et al. [80] (2002) examined the use of research methods (such as action research, conceptual analysis, and data analysis) in 369 papers over a five-year period from 1995 to 1999 across six leading SE journals. Their study adopts a more granular perspective on research techniques, similar to the approach taken in our investigation. The authors conclude that the methodological diversity in SE research remains limited at that

[2] `https://dblp.org/db/conf/cbse` [Last accessed on 2025-08-24]
[3] `https://dblp.org/db/conf/ecsa` [Last accessed on 2025-08-24]
[4] `https://dblp.org/db/conf/icsa` [Last accessed on 2025-08-24]
[5] `https://dblp.org/db/conf/qosa` [Last accessed on 2025-08-24]
[6] `https://dblp.org/db/conf/wicsa` [Last accessed on 2025-08-24]
[7] `https://dblp.org/db/conf/sigsoft` [Last accessed on 2025-08-24]
[8] `https://dblp.org/db/conf/kbse` [Last accessed on 2025-08-24]

time, with a narrow range of approaches being predominantly employed (i.e., conceptual analysis, conceptual analysis /mathematical analysis, and concept implementation).

Additionally, Zelkowitz and Wallace [328] (1997) proposed a twelve-model classification schema for experimentation within the software development domain. Their framework aims to evaluate the extent to which SE papers substantiate their claims through empirical or experimental methods. While their focus is primarily on empirical validation, our study also considers non-empirical evaluation strategies relevant to SWA research objects, thereby broadening the scope of methodological assessment.

(Secondary) Studies on Validity Issues in Software Engineering Research Threats to Validity TtV are a critical concern in empirical SE, directly impacting the credibility and generalizability of research findings. Several studies have investigated how TtV are reported and addressed across SE literature.

Early work by Feldt and Magazinius [64] (2010) revealed that a significant portion of empirical studies either omitted TtV discussions or addressed them superficially. Their survey of 43 papers published at the ESEM[9] conference in 2009 highlighted inconsistencies in terminology and a lack of structured mitigation strategies. Based on this literature analysis, they present, among other things, recommendations on how to better support validity analysis in the future.

Furthermore, Wright et al. [315] (2010) discuss challenges to internal, external, and construct validity in SE research. They investigate, via a literature study of papers published at the ICSE and FSE conference (2007–2009) using real-world software systems, the extent to which external validity concerns are explicitly and adequately addressed. Their analysis highlights the need for more rigorous treatment of validity threats to ensure the generalizability and reliability of empirical findings in the field.

Similarly, Ampatzoglou et al. [8] (2019) conducted a tertiary study on evidence-based research, identifying ad hoc categorization and inconsistent reporting of TtV. They proposed a classification schema and checklist to improve rigor in secondary studies.

In a position paper by Verdecchia et al. [294] (2023), the authors aim to open a discourse on the current use of TtV sections in empirical SE papers. The authors identify key limitations in current TtV practices, including the lack of differentiation among validity types (categorizations) and the tendency to treat threats superficially just as an afterthought. For each issue, they propose improvements aimed at fostering more rigorous and reflective engagement with TtV in the research community.

More recently, Lago et al. [168] (2024) analyzed award-winning (ACM SIGSOFT Distinguished Paper Award) ICSE papers and found that TtV sections are often treated as formalities rather than substantive analyses. Their work emphasizes the need for cultural and procedural shifts in how validity threats are integrated into study design and reporting. They offer twelve actionable recommendations to enhance transparency and methodological robustness.

Collectively, these studies underscore the need for standardized guidelines and community-wide awareness to ensure that TtV are systematically identified, categorized, mitigated,

[9] `https://dblp.org/db/conf/esem` [Last accessed on 2025-08-24]

and discussed. Addressing these issues is essential for advancing evidence-based practices and improving the reliability of SE research.

(Secondary) Studies on Replication Aspects in Software Engineering Research Regarding *replication issues*, we consider the mapping study by Cruz et al. [48] (2020) as related, as the authors conducted a literature study by including papers published from 2013 to 2018 and selecting dedicated venues (137 primary studies) that reported at least one replication of an empirical study in SE. The results indicate that the number of replication studies increased: 1994–2003 (average 4.1 replications/year), 2004–2009 (11.7/year), 2010–2012 (24.3/year), and 2013–2018 (average 22.8 replications/year). Some publications include multiple replications, suggesting that the actual replication activity may be even higher. These findings indicate a clear upward trend in replication efforts within the field during the 2013–2018 period.

Hermann et al. [99] and Winter et al. [308] have contributed to understanding Artifact Evaluation (AE) in SE through two complementary studies. The first study, published in 2020, reveals a disconnect between formal AE guidelines and the expectations of reviewers. It highlights inconsistent terminology, especially around replicability, and proposes clearer standards and better alignment to improve transparency and effectiveness. The second study, published in 2022, analyzes AE practices across ten years of conference publications. It finds that, while AE does not consistently increase citation impact, evaluated artifacts are generally better documented and more reusable. The study identifies tensions between reproducibility and reusability and offers data-driven recommendations to refine AE processes. These studies underscore the need for clearer criteria, standardized terminology, and improved reviewer guidance to enhance the role of AE in promoting reproducible and credible research.

In SWA research, Kazman et al. [143] (2023) address a critical gap: the lack of accessible, curated, and reusable datasets necessary for conducting rigorous empirical studies. While empirical methods are increasingly recognized as essential for advancing knowledge in SE, the authors argue that SWA research still lacks the foundational infrastructure, particularly in terms of data availability. These artifacts are essential for supporting a wide range of empirical studies, from architectural reconstruction to design decision analysis. However, in SWA, datasets are often difficult to obtain, poorly maintained, and rarely reused. The authors propose establishing standardized, FAIR-compliant datasets to support empirical SWA research. These datasets would enable replication and validation, and foster more impactful studies. The paper calls for a collective effort in the research community to build shared data infrastructure, aligning SWA with Open Science and enhancing its credibility, reproducibility, and practical relevance.

Guidelines for Venues and Research Methods / Experience Reports As our work on a classification and data schema for SWA research aims to support researchers with initial knowledge based on how to conduct and assess SWA research, we consider guidelines of conferences or journals (e.g., "How to get your paper accepted at OOPSLA" (1991) [271], "How to write a successful SoSyM submission" (2016) [85]) and SE quality initiatives (cf. The ACM SIGSOFT Empirical Standards [232]) as another related area of our work.

These guidelines focus either on the optimization of papers for a specific venue or on the provision of checklists for research methods in SE. In our approach, we follow a more general and holistic way to describe both research artifacts, namely papers and replication packages, and instantiate our data schema for SWA research.

Also Falessi et al. [61] (see also previous work [62]) map the empirical paradigm with respect to its applicability to SWA research. They state that the major efforts in SWA research to develop new methods, techniques, and tools should be aligned with a rigorous empirical assessment. In an experience report, they describe thirteen challenges and ten lessons learned that they derived from the use of, e.g., controlled and replicated experiments. In conclusion, they support and suggest greater synergy between the empirical SE and SWA communities.

Conclusion In conclusion, we can observe an increase in and special interest in empirical work (i.e., empirical research methods) applied in SWA research. This trend has evolved toward more focused investigations into critical empirical dimensions, such as concerns regarding validity and the challenges of replication. Our work to construct and evaluate a classification and data schema for SWA research based on the general concepts *Findings*, *Validity*, and *Evidence* lies at the intersection of the aforementioned related works (both before and after the publication of our work). By adopting a statement-based approach to describe research artifacts, our schema facilitates a more structured and granular representation of research knowledge that is outstanding in comparison to other related approaches. This, in turn, supports the construction of a comprehensive and reusable knowledge base for the SWA research community.

14.1.4. Unified Approaches in Software Engineering Research

Our fourth contribution constitutes a general research process to construct, benchmark, maintain, and evolve (hierarchical) classification and data schemas over time. Maintenance in this context refers to the correction, adaptation, and improvement of knowledge description concepts in response to community standards and state-of-practice developments. Evolution, by contrast, involves the proactive and strategic modification of these concepts to address normative, long-term research practice goals.

In this section, we introduce and discuss research that is related to our work. First, we regard unification approaches (based on the mapping of different approaches) w.r.t. research classifications and knowledge description approaches in SE as related by analyzing the following criteria: the unification purpose (i.e., why and for what purpose was the unification done?), the unification process (i.e., how was the unification done? What were the steps to achieve the unification of the classification?), and the evaluation (i.e., how was the unified classification evaluated?). Second, we consider additional unification and knowledge description approaches in SE research that employ alternative descriptive techniques, such as processes, guidelines, or checklists. These are also analyzed using the aforementioned criteria to ensure consistency and comparability. Lastly, we consider tertiary studies in SE meta-research and approaches that offer a general research process that can be applied and adapted to similar research objects.

Unification of Research Classifications related to Software Engineering In 2005, Vessey et al. [295] identified limitations in effective knowledge sharing across existing classification systems within the research discipline of computing, which includes research fields like Computer Science (CS), Software Engineering (SE), and Information Systems (IS). Their primary objective was to develop a unified classification schema that accommodates the needs of all three fields, enabling meaningful comparison and contrast across disciplines.

The authors began by reviewing existing computing classification systems, noting that these predominantly emphasize research *topics* while neglecting other dimensions. Through conceptual analysis, they identified key characteristics necessary for a more comprehensive classification and proposed a multi-dimensional schema comprising five research-oriented dimensions: *Topic*, *Research Approach*, *Research Method*, *Unit/Level of Analysis*, and *Reference Discipline*. This proposal aims to foster an understanding of where intersections exist among the three computing disciplines and where they are distinct.

To fill the five dimensions with concrete classes, the authors identified the key features and procedures based on existing literature for creating a classification system that is non-overlapping and non-redundant classification system. A pilot study involving a bottom-up review of existing literature informed refinements to the schema. For instance, they introduced the category 'Not Applicable' within the Reference Discipline dimension to account for papers that either lacked a reference discipline or drew from multiple disciplines without a clear primary source.

Finally, they evaluated the effectiveness of the final, unified classification system. Therefore, they reviewed 1,500 articles (488 in IS, 628 in CS, and 369 in SE) using pairs of coders and iteratively classified papers according to the schema, resolving disagreements in each coding session. According to the authors, they achieved a high level of IAA. However, they did not publish this dataset. The study concludes by emphasizing the need for ongoing research to track the evolution of these disciplines over time.

Unification based on different description approaches Furthermore, we regard other unification efforts in meta-research based on different knowledge description approaches as related to our work. These efforts aim to address the heterogeneity in how research in SE is described, documented, and disseminated. In particular, unification initiatives based on structured description approaches, such as guidelines, checklists, and theoretical frameworks, have emerged as critical instruments for improving the consistency and completeness of empirical reporting.

Jedlitschka et al. [119] (2008) identified a common body of knowledge underlying the heterogeneous reporting styles of experiments in SE. Their study revealed that essential information is frequently difficult to locate or entirely absent from published reports, thereby impeding replication and secondary analysis. To mitigate this issue, the authors compared existing guidelines for reporting controlled and quasi-experiments and proposed a unified framework that consolidates best practices for experimental reporting. This unified framework serves as a reference model for researchers, promoting transparency and facilitating the synthesis of empirical evidence.

Complementing this effort, Wieringa [304] (2012) introduced a unified checklist designed to support the reporting of empirical research in SE more broadly by identifying

commonalities and differences between experimental and case study research. Unlike earlier checklists that focused narrowly on experimental research or case studies, Wieringa's checklist encompasses a wider spectrum of empirical approaches. It provides structured guidance for documenting, e.g., research objectives (e.g., problem investigation), research design choices, and research evaluation.

In a more foundational contribution, Perry [218] (2011) proposed a unifying theoretical framework for SE research. The motivation behind this work is twofold: first, to establish a unified foundation for SE as a scientific discipline; and second, to align empirical SE research with the rigorous methodological foundation and standards observed in the natural and behavioral sciences. Perry's framework emphasizes the importance of theory-driven inquiry and the systematic accumulation of (research) knowledge.

Tertiary Studies in Software Engineering Meta-Research We also consider tertiary reviews (i.e., reviews of secondary studies addressing the same research questions, as defined by Kitchenham and Charters [147]) that address meta-research topics according to Ioannidis et al. [112] and indicate the maturity of a research object within these topics.

Most of these tertiary studies are centered around evidence-based studies. For instance, they investigate a specific type of secondary study, such as systematic literature reviews (Kitchenham et al. [148] (2013)) or systematic mapping studies (Petersen et al. [221] (2015)); the research questions asked in secondary studies (Silva et al. [264] (2011)); or the reporting of TtV in secondary studies (Ampatzoglou et al. [8] (2019).

Unlike these studies, which concentrate on specific concerns within the meta-research area *Methods* (cf. Section 2.1.1), our work adopts a broader perspective. Specifically, we analyze classifications spanning all five meta-research areas as defined by Ioannidis et al. [112], with the goal of providing an integrated framework for characterizing research artifacts such as scholarly papers and replication packages.

General Research Process for Comparability and Reuse In our work, we propose a unified approach and a generalizable research process for the development and evaluation of classification schemas, specifically categorical descriptions related to construction and evaluation concerns (cf. Section 5.3 and Chapter 8). Our approach is inspired by any kind of meta-research efforts that aim to establish (community) standards to investigate or evaluate a specific research object (e.g., classification schemas for SE research). In this doctoral thesis, we provide an approach that goes beyond checklists for research methods and fosters the potential for reuse in the method design by describing methodological patterns. For instance, Winter and Schelp [307] highlight and discuss methodology reuse opportunities and potentials by summarizing the state-of-the-art in reference modeling and method engineering. By adopting such a pattern-oriented perspective, our approach facilitates the comparability of research designs and results across studies that address the same or similar research objects.

Conclusion We can identify existing unification efforts of various knowledge description approaches or bodies of research knowledge. These unification approaches are mostly based on *Argumentation*, sometimes accompanied by tabular forms for comparison and the

mapping of different approaches. While these approaches provide valuable insights, they tend to lack a systematic and transparent foundation for broader long-term applicability. In contrast, our proposed unification strategy is grounded in a general, metric-based, and transparent research process. Initially, our unification approach is important to identify and close research gaps, harmonize similar approaches, and to foster a common and shared understanding of metadata in SE research. Over time, our approach contributes to the maintenance and evolution of the derived and unified classification and data schema. This dynamic adaptability is a distinguishing feature of our approach, enabling continuous refinement and alignment with emerging research practices as well as normative considerations. As a result, our unification and general research approach promotes maturity, comparability, and reusability, thereby contributing to the sustainability of literature studies (cf. Section 6.2) and knowledge description, persistence (populated data at the M1 level, cf. Section 5.2.2), and access.

14.2. Knowledge Persistence and Access

In the following sections, we review related work that is aligned with the individual contributions presented in the second part of this doctoral thesis. The structure mirrors that of the previous section: each subsection is explicitly aligned with a specific contribution, and the order of presentation is consistent with the sequence in which the contributions (related to the overall approach KARAGEN, C5, C6, and C7) were introduced.

14.2.1. Methodological Preliminaries

In this section, we first provide related work on methodological preliminaries. This includes an overview of QA datasets in the scholarly domain and efforts towards guidelines for LLMs in empirical studies in SE. Both are analyzed and put into relation to our approaches.

QA Datasets Targeting the Scholarly Domain Following, we review QA datasets in the scholarly domain and compare them to our constructed and employed dataset in Chapter 13.

The **ORKG-QA Table dataset**[10] (2020) has a `CSV` format and consists of data integrating 13 tables collected from the ORKG infrastructure and derived from information contained in over 100 academic publications, according to the format *question, table (ORKG identifier), type, answer*. In total, the dataset contains 80 question-answer pairs in English. The dataset encompasses diverse question types, including aggregation queries (e.g., minimum, average, and most frequent values), boolean queries (i.e., true/false), list-based queries that require enumerating relevant entities, and queries that rely on the synthesis and combination of information from multiple sources. The question types are balanced as follows: aggregation (16), e.g., "What is the most common Knowledge representation method?" (R8364); ask (5); empty (4); list (3), e.g., "Which papers use datastet [sic] DBLP?" (R6187); multiple (2); normal (31), e.g., "What type of evidence is used by publication "*Name of Publication*"?" (R6187); partial (5); related (9); and similar (5). Consequently, the most frequent question type is “normal” (~39%) addressing individual cells in tables.

[10] `https://data.uni-hannover.de/dataset/orkg-qa-table-dataset` [Last accessed on 2025-07-14]

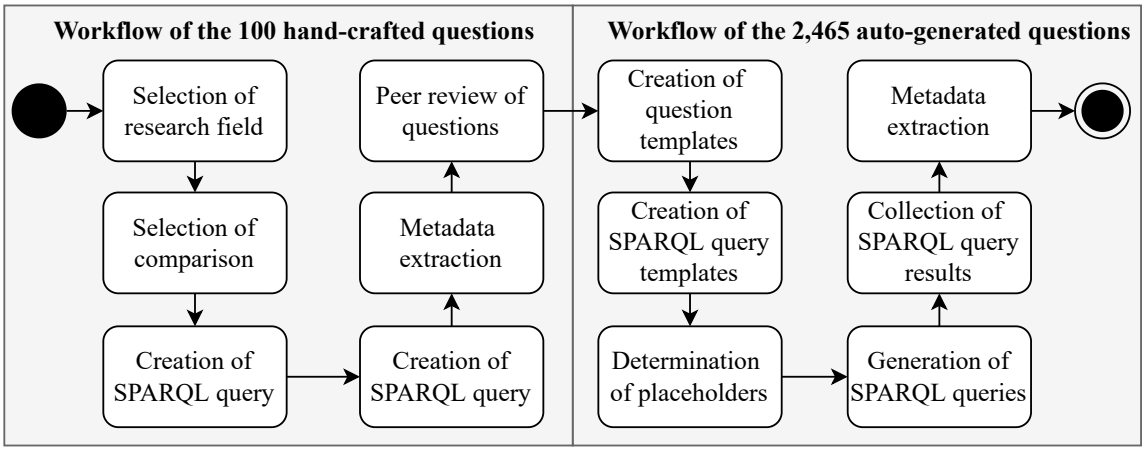

Figure 14.3.: Detailed workflow for the development and generation of the SciQA benchmark according to Auer et al. [12]

The **SciQA dataset**[11] (2023) benchmark consists of a QA dataset and a data dump in NTriples format (RDF NT) taken from the ORKG system. The dataset is provided as `JSON` files and split into test (20%), training (70%), and validation (10%) test examples. SciQA contains 2,565 SPARQL query-question pairs along with answers in English fetched from the ORKG where 100 items are handcrafted (e.g., example question: "What is the most common knowledge representation method in Semantic Representations of Scholarly Communication?") and 2,465 items are auto-generated. The workflow for dataset generation is depicted in Figure 14.3. The questions in the dataset are classified according to the following dimensions: question focus (e.g., Who-What) based on the approach by Moldovan et al. [195]; ORKG content (paper-based or comparison-based); question content (factoid or non-factoid); and the structural properties of the corresponding SPARQL queries, including the number of triple patterns, the query shape, and the presence of specific components such as SELECT, ASK, and COUNT.

The **DBLP-QuAD dataset**[12] (2023) has a `JSON` format and represents a QA dataset over the DBLP scholarly knowledge graph. It consists of 10,000 question-answer pairs, equally divided as creator-focused or publication-focused and grouped into training (70 %), validation (10 %), and test (20 %) sets, with the corresponding SPARQL queries. It can be executed over the dblp KG to fetch the correct answer. The question formation process began with human-written templates, and, later, continued with a machine-generated process forming questions at scale from these templates. The dataset consists of 10 different query types, including Single Fact, Multiple Facts (connection of two or more facts to answer), Boolean (true or false), Negation, Double Negation, Double Intent, Union, Count (count of occurrence of facts), Superlative (min, max) / Comparative, and Disambiguation (e.g., "Which author with the name Li published the paper about Buck power converters?).

The Computer Science Knowledge Graph [53] **CS-KG dataset**[13] (2022) was developed to assess the performance of the automatic methodology employed in constructing the

[11] `https://zenodo.org/records/7744048` [Last accessed on 2025-07-14]
[12] `https://zenodo.org/records/7643971` [Last accessed on 2025-07-14]
[13] `https://scholkg.kmi.open.ac.uk/` [Last accessed on 2025-09-25]

underlying KG. It was manually developed and independently annotated by three senior computer science researchers, comprising 1,200 statements involving subjects and objects classified under Machine Learning subtopics, as defined by the Computer Science Ontology (CSO)[14]. The dataset was assembled by aggregating five distinct subsets: 200 statements with high support (greater than 5), 200 statements with moderate support ($\geq$ 3), 200 statements with low support (<3), 400 statements that were discarded by the automatic methodology, 200 artificially generated statements not present in the KG. Annotators assigned a binary label: 1 for correctness (i.e., the statement is supported by scientific literature) and 0 otherwise. External resources were permitted to verify the validity of each triple. The inter-annotator agreement, measured using Fleiss' κ, was 0.435, indicating a moderate level of consensus among experts (cf. Section 2.8.3).

In contrast to these works, our dataset (in a `CSV` format) for benchmarking RAG approaches in a KGQA setting based on our classification and data schema for SWA research (cf. Section 7.4) and the corresponding labeled dataset, populated in the ORKG, consists of 170 appropriately balanced QA pairs according to our question taxonomy schema with varying degrees of retrieval complexity, as outlined in Section 10.3.

AI / LLM Guidelines Empirical studies involving LLMs, whether they use LLMs as tools, embedding them in SE tools, benchmarking their outputs, or investigating their real-world usage, pose, among other things, novel reproducibility challenges. These challenges stem from factors such as evolving model versions, stochastic behavior, hidden and undocumented configuration parameters, and non-deterministic training settings. To address these concerns, a draft set of guidelines has been developed, informed by discussions at the 2024 International Software Engineering Research Network (ISERN) meeting and the 2nd Copenhagen Symposium on Human-Centered Software Engineering AI. This draft serves as an initial framework to foster further discourse within the research community, with the objective of establishing a shared understanding of best practices for conducting and reporting empirical studies involving LLMs [298].

In contrast to our work [132], we enrich the SE research landscape and the aforementioned guideline by adding the sustainability dimension, i.e., environmental impact in terms of carbon- and energy-efficiency, as a critical evaluation consideration in LLM studies. Besides introducing our novel combined metrics (cf. Section 5.3), we also provide an overview of experiment management tools and concrete best practices for conducting experiments involving LLMs, especially when considering the assessment of energy consumption. Consequently, one can evaluate the complementation of both approaches in future work.

14.2.2. Open Research Knowledge Graph Template Specifications

We are the first to propose and provide ORKG templates for the research field of *Software Architecture and Design*, thereby supporting our corresponding ORKG observatory[15]. Therefore, we provide different variants of our classification and data schema for SWA

[14] `https://cso.kmi.open.ac.uk/home` [Last accessed on 2025-10-03]

[15] `https://orkg.org/observatories/Software_Architecture_Research`, [Last accessed on 2024-01-05]

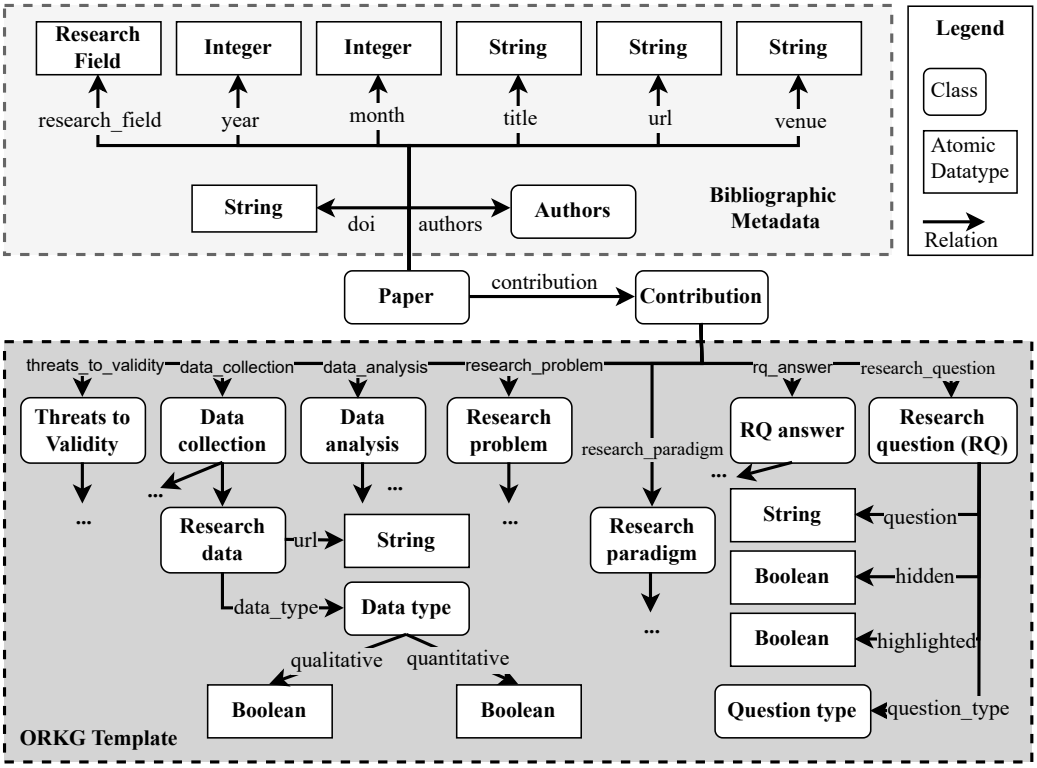

Figure 14.4.: Excerpt from the ORKG template for Requirements Engineering (Figure adapted from Karras et al. [142] and Karras [140])

research (cf. Section 7.4) as ORKG templates to build different knowledge bases to the same version of the schema for our experiments in Chapter 13. This constitutes our fifth contribution. We also highlight the difference between descriptive (i.e., state-of-the-art) and normative (i.e., a desirable state to reflect on and to evolve into this state) aspects.

In this section, we provide an overview of ORKG template (cf. Section 2.6) specifications and corresponding construction processes and evaluation efforts related to (software) engineering research and discuss differences in these procedures and our approach. Both of the following works pursue the research goal specification as defined by the GQM approach (cf. Section 2.8.1). The purpose is to support the organization of FAIR scientific knowledge within the respective research fields. This organization aims to improve communication, reuse, and sustainable literature reviews. It does so from the point of view of domain experts (researchers) in these fields. The context is given by the respective use cases, namely describing empirical practices and forming process chains.

Requirements Engineering (RE): Karras et al. [142] introduce in their paper "Divide and conquer the EmpiRE: A Community-Maintainable Knowledge Graph of Empirical Research in Requirements Engineering" (2023), **KG-EmpiRE**, an ORKG template for Requirements Engineering (RE) to describe empirical research practice in this research field, as depicted in Figure 14.4. Their research approach encompasses the three phases: Data collection, data extraction, and data analysis. For data collection, they include papers published at the IEEE International Requirements Engineering Conference (IEEE RE) from 2000 to 2022, obtaining 570 empirical studies. For data extraction, they first conduct a

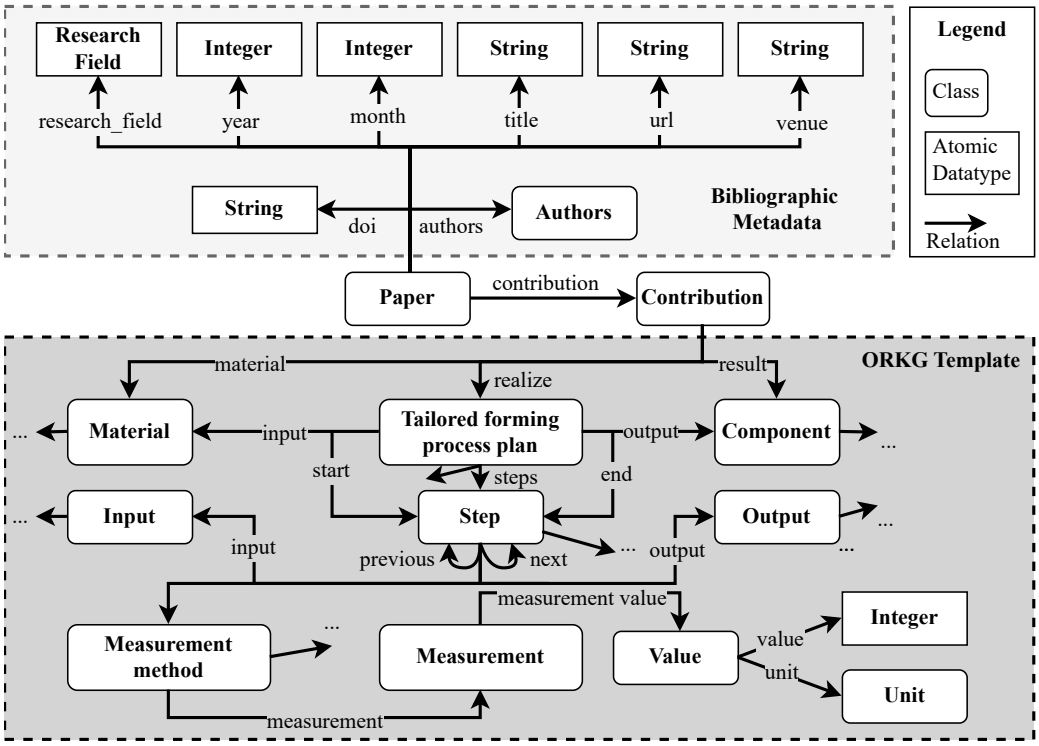

Figure 14.5.: Excerpt from the ORKG template for Engineering Science (Figure adapted from Karras et al. [141])

non-systematic review of 14 publications (published between 2005 and 2021) that provide an overview of the current state and evolution of empirical research in RE and SE by addressing the topic in general and that are not limited to a specific meta-research aspect. Based on this literature data, they encode the key study themes of data collection, research method, bibliographic metadata, data analysis, research paradigm, and research design to derive content types (e.g., threats to validity and research questions, see Figure 14.4) used to build an ORKG template for the data extraction form and populate the RKG.

They evaluated the resulting RKG using 77 competency questions (cf. Section 2.8.8). To this end, corresponding SPARQL queries were constructed and specified, based on an empirical RE vision paper by Sjøberg et al. [267] (2007). Of the 77 questions, 16 were successfully answered, corresponding to approximately 21%. The data analysis revealed both encouraging alignment with the long-term vision and persistent gaps.

Building on this work, Karras [140] expanded and refined KG-EmpiRE in 2024. The updated RKG covers over 680 empirical RE papers, spanning the period from 1994 to 2022.

In combination, these works illustrate a methodological progression: from the proof-of-concept development of a semantically rich RKG of empirical studies to a mature, expansive, community-sustained research infrastructure. They highlight the technical feasibility, community engagement, and value of using ORKG-based knowledge graphs to support FAIR, sustainable, and updatable literature reviews in empirical RE [142, 140] following the principle of *Divide et Impera*, as indicated by the name of the respective RKG.

Engineering Sciences (ES): Karras et al. [141] investigate in their paper "Organizing Scientific Knowledge from Engineering Sciences Using the Open Research Knowledge Graph: The Tailored Forming Process Chain Use Case", how scientific knowledge in engineering science (ES) can be systematically structured, compared, and made both machine-readable and reusable using the concept of ORKG templates. An excerpt of the derived template in this research field is depicted in Figure 14.5.

This use case example focuses on the description and analysis of Tailored Forming Process Chains (TFPCs) as documented in five publications from the Collaborative Research Centre CRC 1153[16], covering a total of ten distinct TFPCs. Their research approach follows the same procedure described in the aforementioned KG-EmpiRE paper by Karras et al. [142] regarding the phases of data collection, data extraction, and data analysis. Further, the paper describes an additional validation phase conducted through a survey with 21 experts in this research field. To this end, the authors developed a reusable, domain-specific ORKG template that allows for the formal representation of key characteristics of the process chains such as process steps, materials used, methods applied, measurement parameters, and outcomes, based on the aforementioned publications in ORKG. Based on this structured data, a comparison artifact was created and published in the ORKG, providing an interactive, tabular overview of all ten TFPCs. As part of the study, domain experts formulated eight competency questions, which are inquiries of practical relevance, typically posed during the analysis of such process chains. These questions address topics such as material combinations or measurable effects of specific processing steps. These questions were answered using SPARQL queries directly executed on the structured ORKG content. To evaluate the classification concept underlying this template, the authors investigated the usefulness of the data extraction topics, their relationships, and the relevance of the knowledge described through an expert consultation with 21 experts.

Conclusion Both approaches showcase the versatility and flexibility of ORKG templates in structuring complex, domain-specific scholarly knowledge. The engineering sciences paper emphasizes expert-driven reuse and construction of process knowledge supporting the description of cycles in the graph, while EmpiRE focuses on community-driven curation and construction of empirical studies (in RE), thereby structuring this research knowledge in a hierarchy. Together, they highlight ORKG's potential to support FAIR and transparent research practices across diverse scientific domains.

In contrast to our approach, we follow a systematic approach by quantifying the construction and evaluation process for hierarchical classification and data schemas, supporting future (descriptive or normative) changes to ORKG template versioning, thereby enabling transparency in the documentation process.

14.2.3. Automated Classification Approaches for Software Engineering Research Artifacts

Our sixth contribution constitutes a classification framework that supports multi-label classification problems comprising the strategies of fine-tuning and prompt engineering as

[16] `https://www.sfb1153.uni-hannover.de/en/` [Last accessed on 2025-08-24]

well as various techniques (which can be differently combined), such as data augmentation or author-splitting in fine-tuning, and the logging of different ML assets. Besides the implementation of the framework, we explored the dataset quality in a retrospective by establishing a categorization schema with labels for structured abstracts. In the following, we present related work on our classification framework and on structured abstracts in SE.

Scientific datasets for text classification We start with an excerpt of different datasets for text classification tasks that address a hierarchical multi-label problem in a scientific context, especially in computer science.

The WOS dataset[17], introduced by Kowsari et al. [157] (2017), comprises three variants (WOS-11967, WOS-46985, and WOS-5736) and is derived from the Web of Science corpus. The WOS-11967 dataset contains 11,967 documents with 35 categories, which include 7 parent categories. The WOS-46985 dataset contains 46,985 documents with 134 categories, which include 7 parent categories. The WOS-5736 dataset contains 5,736 documents with 11 categories, which include 3 parent categories.

The AAPD (ArXiv Academic Paper Dataset) dataset[18], introduced by Yang et al. [319] (2018), contains the abstract and the corresponding subjects of 55,840 papers in the computer science field from ArXiv[19]. A paper may have multiple labels, although the hierarchy is relatively shallow. There are 54 labels from ArXiv subject areas.

The Microsoft Academic Graph (MAG[20], retired on Dec. 31, 2021) dataset, released by Cohan et al. [44] (2020), contains fields of study, authors, institutions, journals, and conferences as labels. MAG offers a large-scale heterogeneous graph of academic entities. While not explicitly designed for hierarchical multi-label text classification, subsets of MAG can be extracted to support hierarchical classification based on fields of study.

The SciHTC dataset[21], introduced by Sadat and Caragea (2022) [246], is one of the largest publicly available resources for hierarchical multi-label text classification. It comprises 186,160 scientific papers sourced from the ACM Digital Library and annotated using the ACM Computing Classification System (cf. Section 3.1.4), which includes 1,233 hierarchical categories (13 nodes at tree level 1). Each document contains a title, abstract, and author-specified keywords, making it suitable for both flat and hierarchical classification models.

The WOS-HMLTC dataset and its variants WOS_JT (43,366 total samples), WOS_CT (65,200 total samples), and WOS_JTF (42,926 total samples) by Toit et al. [286] (2024), were created from Web of Science title and abstract data categorized into a hierarchical, multi-label class structure. The WOS_JT dataset uses the journal-based (JT) classifications as labels. The WOS_CT dataset only uses the citation-based (CT) classifications as labels. The WOS_JTF dataset uses the JT Filtered (JTF) classification schema that filters out documents and classes that do not have a clear overlap between JT and CT classifications.

Our constructed dataset (2021/2022) [155] derived from a literature study (cf. Section 7.4) contains domain-specific content w.r.t. the description of SWA research practices from

[17] `https://data.mendeley.com/datasets/9rw3vkcfy4/6` [Last accessed on 2025-09-25]
[18] `https://github.com/lancopku/SGM/` [Last accessed on 2025-09-25]
[19] `https://arxiv.org/` [Last accessed on 2025-09-25]
[20] `https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/` [Last accessed on 2025-09-25]
[21] `https://github.com/msadat3/SciHTC` [Last accessed on 2025-09-25]

the ECSA and ICSA conference series, represents a (partly hierarchical) multi-label classification problem, and is highly imbalanced with missing representatives, as analyzed in Section 10.4.2. This dataset covers the main challenges for an automated approach, as discussed in Section 12.2. To the best of our knowledge, this is the first dataset for SWA research and artifact content-related metadata, such as the description of findings and validity, that goes beyond topic classification compared to the previously introduced datasets.

Automated Classification Approaches in SE Research First, we regard domain-specific fine-tuning (transfer learning) of general-purpose models like BERT [54] as related to our framework. In a scientific context, we can find examples such as SciBERT[22] [25] (pretrained on 1.14 million full-text scientific papers from Semantic Scholar, covering domains like computer science (~18%) and biomedicine (~82%)), BioBERT [173], SciDeBERTa [121], Galactica [281] (online for just three days[23]), and ScholarBERT [104]. A recent comparative study (preprint) by Rostam and Kertész [238] (Sept. 2025) conducted a systematic evaluation of domain-specific and general-purpose pre-trained language models (PLMs) for classifying scientific texts. The authors use the WOS dataset. Their results show that domain-specific models (e.g., SciBERT and BioBERT) outperform general-purpose models in both abstract-based and keyword-based classification tasks, especially in specialized scientific fields, highlighting the value of domain adaptation in scientific NLP tasks. This study underscores the importance of our developed classification framework, which enables the capability to support more complex scientific document classification tasks with domain-specific adaptations, majorly enhancing state-of-the-art studies and practices.

Second, we consider applying (advanced) prompting techniques, e.g., in-context learning (ICL), to classify scientific artifacts as related to our framework. For instance, Wang et al. [299] (2023) present a knowledge-guided prompt learning framework that integrates conceptual knowledge into prompt templates. This method enhances few-shot classification performance by aligning implicit model knowledge with external domain-specific information. They evaluate their approach on three Chinese text classification datasets. Kunnath et al. [166] (2023) investigate the effectiveness of various prompting strategies for citation classification (i.e., flat classification of background, compares contrasts, extension, future, motivation, and uses). The authors evaluate multiple techniques including zero-shot and few-shot prompting, prompt ensembling, chain-of-thought prompting, and activation fine-tuning on legal text datasets. Their findings reveal that while prompt engineering can enhance LLM performance on general tasks, its effectiveness does not consistently transfer to domain-specific contexts such as legal citation classification. The complexity of legal language and document structure poses challenges that limit the generalizability of standard prompting methods. The study highlights the need for domain-adapted prompting strategies to achieve reliable classification in specialized fields. SciPrompt by You et al. [324] (2024) introduces a knowledge-augmented prompting framework for the fine-grained classification of scientific topics. The method automatically enriches verbalizers by

[22] `https://github.com/allenai/scibert` [Last accessed on 2025-09-25]

[23] `https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/` [Last accessed on 2025-09-25]

retrieving semantically correlated domain-specific terms, weighted by correlation scores to improve prediction accuracy. SciPrompt outperforms baseline prompt-based methods in few- and zero-shot settings, particularly for emerging scientific topics. Wang et al. [300] (2024) propose a meta-knowledge prompt engineering approach for generating research questions from scientific literature. By embedding domain-specific features into prompts, their method improves the extraction of research objectives and methods, achieving high F1 scores across multiple models. Maiti et al. [189] (2025) conduct a comparative analysis of GPT-4o and DeepSeek R1 for scientific sentence categorization. Using prompt engineering, they evaluate model consistency and effectiveness across a curated dataset of scientific papers. Their findings underscore the importance of prompt design in achieving reliable classification outcomes.

In conclusion, the convergence of different classification paradigms such as fine-tuning models and prompt engineering within our open-access, feature-rich classification framework with a modular and extensible design represents a major step toward democratizing access to and development of advanced NLP tools for scientific document classification in SE research and beyond. By fostering collaboration, transparency, domain-specific adaptation (to different hierarchical schemas), and comparability of evaluations results, such a classification framework can catalyze progress across a wide range of scientific disciplines. Moreover, it promotes the tracking of energy consumption and carbon emissions, enabling trade-off decision using our combined sustainability metrics and thereby promoting the concepts of Green AI (i.e., evaluation goal *applicability* with the evaluation strategy *Automated* in our evaluation approach (C1)). In addition, the framework allows for the adaptation or creation of different hierarchies and concepts w.r.t. the data schema and the management of prompting techniques (cf. Section 2.3.1). In the sense of meta-research, it supports Open Science principles, the unification of research practices, and prevents redundant work and duplicate results.

Structured Abstracts in SE Structured abstracts are a standardized format for summarizing scientific papers. Unlike traditional abstracts, which are written as a single paragraph, structured abstracts are divided into clearly labeled sections that reflect the logical flow of the research. Empirical studies in SE show that the information contents of structured abstracts are clearer and more complete (cf. Budgen et al. [35] and Kitchenham et al. [149]). In SE, the use of structured abstracts has gained increasing attention in the literature and is now a submission requirement at some SE venues.

The IMRAD / IMRaD[24] format (Introduction, Methods, Results, and Discussion) is the prominent standard for a structured form of an original research paper and has also been adapted in EBSE and in SE research in general. For instance, the EBSE community at the Durham University[25] recommends five headings (labels) for structured abstracts to describe a range of software engineering topics, namely, Background, Aims, Method, Results, and Conclusions. Further, Jedlitschka et al. [120] (2005) also proposed the use of these headings with additional names for the labels treated as synonyms for structured abstracts in SE: Background or Context, Objective or Aim, Method, Results, and Conclusion.

[24] `https://www.nlm.nih.gov/bsd/policy/structured_abstracts.html` [Last accessed on 2025-08-18]
[25] `https://ebse.webspace.durham.ac.uk/structured-abstracts/` [Last accessed on 2025-08-21]

Similarly, Wohlin [312] (2014) refers to the journal Information and Software Technology (IST), which employs the same five categories: Context, Objective, Method, Results, and Conclusion. More recently, Prechelt et al. [225] (2025) introduced a refined set of labels for classifying abstract components in SE research. These include: Background/Context, Gap, Objective, Design, Method, Result, Summary, Conclusion, and Outlook. Based on their empirical study, the authors also provide recommendations[26] for improving abstract writing practices in the SE community.

Based on these works, we can observe a constant interest in discussing and defining appropriate labels for structured abstracts that can accommodate the diverse range of research paradigms within SE and enable rapid comprehension of scientific papers.

In addition to these approaches, we propose a set of structured abstract labels that explicitly integrate principles of FAIR and Open Science (cf. Section 2.1.3). For instance, the inclusion of a label such as *Replication Package* contributes to a more holistic and transparent overview of research contributions. Moreover, we support our general upper-level concept of specifying papers according to their findings (which advance the state-of-the-art), and corresponding validity and evidence, as introduced in Section 7.4.

14.2.4. Retrieval Approaches

In this section, we review four key areas of related work in KGQA retrieval and scholarly QA systems, each presenting conceptually relevant approaches and highlighting their differences from our proposed HubLink retriever (cf. Section 13.2).

Scholarly QA Approaches Recent efforts in scholarly QA have explored the integration of LLMs with RKGs primarily through semantic parsing methods, translating natural language into formal query languages (e.g., SPARQL), and through RAG using documents and vector stores (e.g., ORKG Ask[27], Consensus[28], Elicit[29] [36], and SciSpace[30] [241, 116]).

Jaradeh et al. [118] propose **JarvisQA** that transforms tabular ORKG data (e.g., surveys, comparisons, or results) into triples, textualizes them using an LLM, and retrieves answers with a BERT-based model [54]. Thus, this scholarly QA system operates only on structured tabular inputs rather than performing direct reasoning over a RKG.

To construct the **DBLP-QuAD** dataset and to build a semantic parsing baseline, Banerjee et al. [18] fine-tunes a Text-to-Text Transfer Transformer (T5) model [229] to translate questions into SPARQL queries, depending heavily on large annotated datasets and accurate entity linking; resources that are often lacking for newly constructed RKGs.

KGMistral [181] employs a RAG system in which predefined SPARQL templates are filled and executed to retrieve relevant triples. These are then textualized and used as context for the LLM. KGMistral demonstrates poor to moderate performance in controlled environments (BLEU 0.008 and Rouge-1 F_1 0.150). Its reliance on static template-based

[26] `https://github.com/serqco/qabstracts/blob/main/serqco-abstracts-structure.md` [Last accessed on 2025-08-18]
[27] `https://ask.orkg.org/` [Last accessed on 2025-07-14]
[28] Consensus: AI Search Engine for Research: `https://consensus.app/` [Last accessed on 2025-07-14]
[29] Elicit: The AI Research Assistant: `https://elicit.com/` [Last accessed on 2025-07-14]
[30] SciSpace AI Research Agent: `https://scispace.com/` [Last accessed on 2025-07-14]

retrieval mechanism, limited generalization capabilities, and scalability challenges underscore the need for more flexible and adaptive QA systems for RKGs.

In a related strategy, Taffa et al. [279] propose a few-shot LLM prompting via Vicuna-13B[31] as a retrieval-based method that reuses SPARQL queries associated with the most similar question in the training set. Although efficient in controlled environments and evaluated in the SciQA dataset, this template reuse limits generalization to novel queries beyond the training distribution. It also lacks schema awareness, making it prone to generating semantically incorrect SPARQL queries. The use of template-based, auto-generated questions in the evaluation dataset may lead to overfitting. Additionally, the method struggles to scale with larger ontologies, and its poor zero-shot performance reduces its applicability in low-resource or dynamic domains.

Evaluation of these systems has often relied on the SciQA dataset, where strong reported performance may be exaggerated; thus, LLMs can potentially exploit recurring patterns [175]. Furthermore, many test questions share schema elements with training data [125], reducing the challenge of schema generalization. This limitation is amplified by the fact that semantic parsing approaches tend to degrade on large, dynamic KGs containing unseen schema elements or entities [247], and they remain dependent on task-specific training data, which hinders scalability.

To address these constraints, prompt-based and schema-agnostic semantic parsing methods have been proposed. Lehmann et al. [175] show that zero-shot prompting performs poorly on SciQA, while few-shot prompting can match fine-tuning in accuracy and improve adaptability, though it still requires examples that do not transfer well to unseen data. Jiang et al. [125] introduce a two-stage prompting framework incorporating ontology information into both structural and content-level prompts, achieving strong results but encountering scalability issues when applied to larger ontologies.

ORKG Ask[32] is a QA system designed to facilitate semantic search and targeted information extraction over scholarly literature and developed by a research community. This naive RAG system integrates vector-based retrieval with LLM processing to deliver contextually relevant answers. User queries and research articles based on title and abstracts, sourced from the **CORE dataset**[33] of open-access publications, are encoded into high-dimensional embeddings via the Nomic embedding model[34]. These embeddings are stored and searched via an Approximate Nearest Neighbor (ANN) algorithm in the Qdrant vector database[35]. Relevant documents are subsequently processed by the Mistral Instruct 7B v0.2[36] LLM, which extracts and synthesizes relevant information.

In contrast to these approaches, our retriever HubLink operates on RDF-based RKGs and embeds subgraphs (hubs) rooted at specific nodes and transforms paths to text, enabling schema-agnostic knowledge adaptation without annotated training data and resource-level linking (by providing DOIs to generated answers) during inference time.

[31] `https://lmsys.org/blog/2023-03-30-vicuna/` [Last accessed on 2025-07-14]
[32] `https://ask.orkg.org/` [Last accessed on 2025-07-14]
[33] `https://core.ac.uk/services/dataset` [Last accessed on 2025-07-14]
[34] `https://huggingface.co/nomic-ai/nomic-embed-text-v1.5` [Last accessed on 2025-07-14]
[35] `https://qdrant.tech/` [Last accessed on 2025-07-14]
[36] `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2` [Last accessed on 2025-07-14]

LLM-Guided Stepwise KGQA Approaches Many training-free KGQA methods adopt stepwise reasoning, decomposing complex queries into manageable steps guided by pre-trained LLMs. These approaches typically involve dynamic traversal of the KG, with the LLM acting as a controller to select entities, evaluate paths, and refine context iteratively. Examples include Think-on-Graph (ToG) [275], Knowledge Solver (KSL) [65], Observation-Driven Agent (ODA) [276], and GRAPH-COT [126], which use strategies like beam search, decision prompts, and chain-of-thought reasoning. Hybrid models such as ToG-2 [187] combine KG traversal with unstructured text retrieval, while Generate-on-Graph (GoG) [318] addresses KG incompleteness by generating plausible triples. The baseline FiDeLiS by Sui et al. [274] integrates a Path-RAG component for retrieving relevant entities and relations with a deductive verification beam-search mechanism, wherein the language model iteratively constructs and validates multi-hop reasoning paths.

In contrast, HubLink retrieves precomputed, semantically anchored subgraphs (hubs) rather than constructing paths dynamically. This design enables faster retrieval and richer semantic context, allowing transitive relations to be incorporated without stepwise traversal. Unlike dynamic methods, HubLink's reasoning operates over structured, embedded hubs, offering a stable and interpretable foundation for LLM-based inference.

KGQA Using Contextual Subgraph Construction Some training-free KGQA approaches involve dynamically constructing subgraphs during retrieval to provide contextual input for LLMs. These methods typically identify entities in the query and expand the graph locally to form evidence subgraphs. For instance, the baseline MindMap by Wen et al. [302] uses prompt-guided LLMs to build path- and neighbor-based subgraphs for structured reasoning. KG-GPT [146] decomposes questions into triples, retrieves candidate relations, and constructs subgraphs for LLM inference. Further, Reasoning on Efficient Knowledge Paths (RoK) [124] applies chain-of-thought prompting to extract entities, generate multi-hop paths, rank them via PageRank, and forward the resulting subgraph to the LLM.

In contrast, HubLink precomputes semantically coherent subgraphs (hubs) offline, anchored on predefined node types (entities). These hubs capture structured relationships and enable fast retrieval, ensuring semantic consistency. Unlike dynamic methods, HubLink does not rely on query-time graph expansion, thereby offering more stable and interpretable context for LLM-based reasoning.

Utilizing Vector Representations for KGQA The fourth area of related work for KGQA systems concerns leveraging dense vector representations to encode knowledge and support retrieval. These methods differ in their embedding strategies and training requirements.

First, we consider approaches that directly embed KG triples using pre-trained models without KG-specific training. For instance, the baseline Direct Fact Retrieval (DiFaR) by Baek et al. [17] encodes individual triples (head entity, relation, tail entity) into dense vectors offline and retrieves relevant facts via nearest-neighbor search between embedded (user) questions and triple embeddings at retrieval time. While efficient for handling simple queries, this method may overlook broader relational context captured in multi-hop paths or larger and complex graph structures due to its focus on isolated triples.

In contrast, HubLink embeds precomputed subgraph structures (Hubs) enriched with textual descriptions derived from paths (with an anchor entity each), triples, and entities. This enables the capture of more complex semantic relationships without requiring dedicated training, while accepting increased storage requirements for indexing.

Second, we regard approaches that involve training of specialized Knowledge Graph Embedding (KGE) models. These methods learn entity and relation embeddings from KG structure. Examples include Pretrain-KGE by Zhang et al. [333], which fine-tunes BERT on KG triples, and KEPLER by Wang et al. [301], which combines KGE with masked language modeling. Algebraic models, such as quaternion and dihedron embeddings [199], have also been explored.

In summary, these approaches demand substantial training data and computational resources. HubLink avoids these efforts and constraints in our preliminary evaluation scenarios by utilizing general-purpose, pre-trained embedding models to encode textual descriptions, enabling flexible application across KGs schemas without additional training, while still providing semantically rich context for downstream reasoning.

Conclusion In general, there are several possible approaches for modern knowledge retrieval techniques (cf. Section 10.1). The rapid advancement of LLMs in recent years has transformed the landscape of information processing and has shown great performance across various NLP tasks, such as question answering. These models operate by leveraging vast pre-trained corpora to generate fluent and contextually relevant, coherent outputs. However, their reliance on internal representations without direct access to external sources introduces limitations, especially the risk of hallucinations and, thus, the lack of factual knowledge which is an essential requirement for scientific retrieval systems.

To mitigate this (and other) issue(s) (as discussed in Section 10.1), RAG was introduced as a hybrid approach (cf. Section 2.3.2) that combines generative modeling with external document retrieval from external knowledge bases. In its naive implementation, RAG utilizes embedding-based similarity search over chunked text segments stored in a vector database (cf. ORKG Ask), improving factual accuracy by grounding responses in retrieved content. Despite its simplicity and effectiveness, naive RAG is constrained by chunking sensitivity / strategies, limited semantic depth, and the inability to perform multi-hop reasoning across scientific documents. However, by integrating RKGs into the retrieval pipeline, graph-based RAG enables structured and semantically rich access to scientific information. KGs encode entities and their relationships, facilitating multi-hop reasoning and contextual disambiguation. This allows graph-based RAG to outperform naive RAG in tasks requiring deep understanding, such as cross-document synthesis, entity linking, and domain-specific QA. Current approaches applied in scientific context, relying on Semantic Parsing, are struggling with schema dependency and require (extensive) training data. In contrast to these approaches, we proposed HubLink, a schema-agnostic (i.e., handling schema changes and evolution, as outlined in Chapter 8), training-free approach that leverages RKGs for scalable and effective scholarly knowledge access.

To foster progress in this research direction, we implemented a modular and configurable retrieval framework, enabling, among others, benchmarking, collaboration, and reuse.

15. Conclusion

In this doctoral thesis, we presented conceptual and technical contributions to enable sustainable and evolvable knowledge representation and access to research artifacts in Software Engineering (SE), with a particular focus on Software Architecture (SWA) research. This chapter concludes the thesis with a presentation of the constructive contributions and the insights gained from our research studies. Therefore, we summarize answers to each main research question in Section 15.1. Further, we discuss the relevance and implications of our contributions for various stakeholders involved in the RLC and the scientific ecosystem, e.g., researchers and practitioners, in Section 15.2. Finally, an outlook on future work that builds upon the presented contributions closes this chapter in Section 15.3.

15.1. Summary

In the following, we summarize results and insights w.r.t. the main research questions (cf. Section 1.4) and their corresponding validation questions (see the respective sections of the main contributions in Part II and Part III).

RQ I-1: Evaluation Approach for Classification Schemas and Its Application Research is a challenging task [4]. Finding adequate methods, research and validation questions, research goals for the method design, and dedicated research steps to evaluate a specific *Research Object* is key for the acceptance in the communities (i.e., research and industry) as well as for the scientific, technical, and societal progress. We identified a lack of methodological and operational guidelines for classification and data schemas as research objects (cf. Section 14.1.1), which are crucial as a basic step for any knowledge description approach (via meta-models, ontologies, knowledge graphs, etc.) and its applications (e.g., knowledge access). Therefore, we developed a three-step sequential method (Section 5.3), where we use the GQM approach (see Section 2.8.1) while dividing the evaluation properties over the research goals: *structure's suitability*, *applicability*, and *purpose*. We built upon our original publication (Kaplan et al. [136]) by adding evaluation strategies for each research goal to further align the underlying research method, highlighting the broad spectrum of evaluation settings and strategies (Kaplan et al. [132] and Kaplan et al. [134]). In this way, we also enable possible future extensions if the research practice evolves. Therefore, we add the internal and external strategy for evaluating the *structure's suitability*, the manual and automated strategy for *applicability*, and, finally, the intrinsic and extrinsic strategy for *purpose*. Regarding the metrics, we consider well-established standard metrics as well as newly proposed metrics. The latter especially refers to metrics for trade-off considerations between performance and carbon emission aspects in AI solutions, for innovation (covering normative aspects) and adaptation (covering descriptive aspects). To

operationalize the method design, we also formulated template questions, embedded in our proposed GQM plan. We presented the impact of our approach via a citation study by analyzing the usage of our proposed solution. Furthermore, we showed the feasibility of our approach, as we provide a running example and applied the method for evaluation in this doctoral thesis for the knowledge description and access approaches in Part II and Part III. In summary, we built a systematic evaluation approach comprising dedicated research goals for the research object *classification and data schema* as an inherent part of knowledge description approaches such as taxonomies or knowledge graphs. Aligned with these goals, we specify properties to investigate, define corresponding template validation questions, and derive metrics to answer these questions, as the GQM approach intends. The main difference from the GQM approach is that we define further dimensions orthogonal to the conceptual (goal), operational (question), and quantitative (metrics) level structure, namely the *evaluation strategy*. This way, we build a conceptual and powerful tool to design the research approach and different stages of assessment maturity in a holistic way beyond checklists for research methods. Moreover, this evaluation approach provides structural flexibility for future extensions, including the incorporation of additional properties, the refinement of template research and validation questions, and the enhancement of associated metrics and evaluation strategies. For instance, additional metrics could be incorporated into the evaluation goal *purpose* w.r.t. the extrinsic evaluation strategy, extending beyond traditional retrieval and generation metrics as well as LLM-as-a-Judge approaches. This may include Human-in-the-Loop aspects, particularly if research practices and standards evolve in the future. In summary, we provided a structured, multi-level evaluation approach for classification and data schemas to enhance guidance and comprehensibility for researchers (e.g., authors and reviewers).

RQ I-2: Enhanced and Tool-supported Documentation Process for Evidence-based Studies
Evidence-based Software Engineering (EBSE) is a methodology, introduced in 2004, that applies empirical research findings to inform and improve SE research and practices. Inspired by evidence-based medicine, EBSE aims to integrate the best available evidence with practical experience and stakeholder values in decision-making processes and research findings related to SE research topics (i.e., core topics w.r.t. the SE development life cycle, education and meta-research aspects). We identified a lack of practical, tool-supported guidelines for conducting sustainable literature reviews that focus on ensuring the long-term persistence and accessibility of research findings (Section 14.1.2). Therefore, we developed a tool-supported process for the data extraction phase in literature studies to support transparent documentation of extracted metadata and facilitate the management of large-scale reviewer collaboration during the extraction. In this way, we were able to enhance well-established, conceptual guidelines for planning, conducting, and reporting literature studies (cf. Kitchenham and Charters [147]). Our practical guidelines use core features from GitLab. This includes branching to enable version control tracking across multiple reviewers, GitLab Wiki to create collaborative documents as an instruction knowledge base, issue (board) and To-Do lists to manage the data extraction process and provide sequential instructions for the conduction of the data extraction task. To operate on machine-readable formats and to manage metadata schemas, we used the SLR Toolkit.

We show the feasibility of our approach in a literature review use case in the context of our proposed classification and data schema for SWA research (C3) by publishing the open-access repository[1] and GitLab Wiki[2] in our replication package [155]. We further used this approach for the documentation of our unified approach (C4). During these studies, we identified optimization potential for the specification of the machine-readable formats, as we faced limitations of the SLR Toolkit in documenting further metadata of the metadata or comments for comprehensibility, which we aim to address in future work (see Section 15.3). In summary, we enhance current conceptual guidelines for literature studies with practical and operationalized support, promoting Open Science and FAIR principles (Section 2.1.3).

RQ I-3: Classification and Data Schema for SWA Research as Knowledge Description Approach
Similar to SE research, SWA research has gained maturity and has experienced an increase in empirical research. We identified a research gap in the description and a missing overview of current research practices in this research field (Section 14.1.3). Therefore, we constructed and iteratively improved a classification and data schema [153] based on our previous work (Kaplan et al. [139, 138], focusing on generalizable considerations to describe research artifacts according to their *Findings* (enhancement of state-of-the-art), the underlying *Validity* (reference to Threats to Validity, describing potential weaknesses or biases in a study that may compromise the credibility and generalizability of its findings) and *Evidence* (reference to data and software that support the statement of findings). We instantiated this schema for SWA research and used parts of our evaluation method (C1) for construction and evaluation. Based on our schema, we conducted a literature review by classifying 153 full technical papers published at ECSA and ICSA (2017–2021), the flagship conferences in SWA research, to describe state-of-practice in this research field. The categorization of the papers and replication artifacts of the included papers were each done by two independent reviewers using our tool-supported documentation guideline (C2). The results show the feasibility of our approach, as the reviewers could resolve disagreements in the post-phase of the (individual and independently conducted) data extraction. The literature results also show that the research field is quite diverse w.r.t. the investigated *Research Objects*, with a total of 13 distinct research objects, with a focus on architecture analysis methods and architecture design methods. The predominant approach to evaluating SWA research objects involves assessing their functional suitability and performance through the application of technical experiments and case studies. With respect to the reporting of threats to validity, our analysis revealed that 97 out of 153 papers (~63.4%) explicitly declared threats to validity. Approximately 17% of the reviewed papers referenced evaluation guidelines, while 14% cited guidelines addressing threats to validity. In total, 17 papers (~11.1%) made their replication package available. Based on these insights, we derived five proposals to foster an improved and informed research practice for the SWA community, highlighting the importance of meta-research and the translational framework (Section 2.1.1). Furthermore, we developed the lookup service

[1] `https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice` [Last accessed on 2025-08-22]
[2] `https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice/-/wikis` [Last accessed on 2025-08-22]

Visulite, supporting an interactive interface for our replication package, as presented in Section 11.4. Notably, the 2026 Open Science Challenge at REFSQ partially adopted elements of our classification schema including the incorporation of the *Evaluation Method* and an adapted version of the *Paper Class* category for paper annotation. In summary, we set a metadata-based standard for general purposes and instantiated it for the research field of SWA research.

RQ I-4: General Research Process and Maintenance Approach for SE Research Classification and data schemas are essential to describe, communicate, and access a body of knowledge. In a scientific context, they contribute to the structured specification of research knowledge regarding research artifacts as metadata concepts. This enhances machine-readability and, thus, enables machine-actionability and allows automated processing, interpretation, and execution by machines without requiring human intervention. Further, the integration of the classification concept and schema in a KG enables AI-readiness [237] (grounded and implemented in C7). However, classification and data schema concepts are predetermined to change over time. It is helpful from a sustainable perspective to be able to track, understand, monitor, and interpret these changes over time. Therefore, we derived a general documentation and research process (Kaplan et al. [137]) based on our evaluation approach (C1) and documentation process (C2) with their incremental and iterative characteristics to track changes metrically. To enhance and enrich our general classification and data schema (C3) with further knowledge statements (besides the core concepts of a paper's findings, validity, and evidence), we conducted a literature study of previous classifications in SE that were used, proposed, or extended. We analyzed the commonalities and differences of the identified classifications and incorporated the concepts into our initial classification schema. In summary, we developed a general research process to evaluate (hierarchical) classification and data schemas that are subject to changes over time.

RQ II-1: Semantic Structuring and ORKG Template for SWA Research Machine-readability of research artifacts is crucial for machine-actionability and for the integration of modern AI techniques in current research applications, such as scientific knowledge access and discovery. Therefore, we are the first to explore the capabilities of ORKG for SWA research (Section 14.2.2) to build a knowledge base (Kaplan et al. [130]) for our retrieval approach (C7). We use our classification and data schema for SWA research [153] to derive different ORKG template variants (Section 11.3). In this process, we face two issues: First, we found discrepancies in state-of-practice and normative aspects when modeling research knowledge based on knowledge statements rather than on the entire document (paper). Modeling knowledge statements based on *Findings*, *Validity*, and *Evidence* requires a strict alignment to not lose semantic richness and adequate abstraction of granularity. However, this strict alignment is not consistent in the papers, e.g., in cases where the paper investigates two or more research objects or when more than one evaluation method is applied. This is also indicated by our study and insights from the reviewed literature in Chapter 7. Notably, a structured semantic knowledge description approach can reveal those issues and increase awareness of them within the research community. This can also improve the quality of assessments in the peer-review process, as the approach contributes to greater comprehen-

sibility. Second, we have to consider trade-off decisions between human-understandability (ease of use) and human-applicability as well as machine-readability, as indicated by our evaluation approach (Section 5.3). For the final ORKG template, we took all these issues into account in order to reflect state-of-the-practice and draw attention to normative aspects. In order to better understand and evaluate the quality of our retrieval approach, we created three additional ORKG template variants based on the main variant. Therefore, we constructed and populated four RKG variants using ORKG as infrastructure based on two orthogonal characteristics: path length (determines the semantic depth of graph traversals) and contribution granularity (affects how research contributions are represented). Finally, we implemented the *Visulite*[3] tool (Kaplan et al. [130]) to better visualize the results of our replication package for researchers and to provide a knowledge management and lookup service for the research community. In summary, we explored the capabilities of ORKG to build a knowledge base for our RAG approach and showed its feasibility in a use case example for SWA research, supporting the sustainable organization of FAIR scientific knowledge to take a first step towards standardization in this field.

RQ II-2: Classification Framework Manual KG population and updating (sustainable) literature reviews are labor-intensive processes, particularly in the light of the rapid expansion of research knowledge. These tasks demand substantial human effort and time. To semi-automate these tasks, we face various challenges that arise from the high complexity of the metadata for research artifacts, as discussed in Section 12.2. These challenges are not collectively addressed in the state-of-the-art (cf. Section 14.2.3). To cope with these challenges, we developed a classification framework for multi-label problems (Kaplan et al. [131, 132]) using the *Blackboard* design pattern to benchmark and derive LLM-based classifiers, supporting both fine-tuning and prompting techniques as well as various additional features (cf. Section 12.3), that can be further integrated into real-world applications. For evaluation and benchmarking purposes, we integrated the performance and (combined) sustainability metrics [132] of the research goal *applicability* embedded in the automated evaluation strategy to support this GQM fragment, as presented in Section 5.3. We conducted an extensive evaluation to show the feasibility of our various implemented features in the classification framework using the classification and data schema for SWA research (Section 7.4) and the corresponding dataset derived from our literature review in a multi-reviewer data extraction context (cf. Chapter 7). As we obtained poor to moderate performance results, although they were still better than random guessing, we further examined the dataset quality by analyzing the information content of the abstracts. Therefore, we first defined labels for sentence classification with the focus on the construction of a structured abstract. Then, we classified the sentences of each abstract in the dataset according to our derived schema. We identified poor information content in the abstracts regarding our schema for SWA research, explaining our poor to moderate / good obtained classification results in this regard. For this, we recommend structured abstracts for the ECSA and ICSA paper submissions to improve AI readiness and support as well as to foster the understandability of research work in the review process and in the post-publication phase, as also indicated in empirical studies (cf. Section 14.2.3).

[3] `https://softwarearchitectureresearch.github.io/visulite/` [Last accessed on 2025-08-22]

RQ II-3: Retrieval Framework and Knowledge Access using Knowledge Graphs and Large Language Models Scientific literature discovery is still remaining document-centric and keyword-based (cf. Section 3.2.2). Current state-of-the-art KGQA approaches in a scholarly setting struggle with evolving schemas and often require training data (Section 14.2.4). Consequently, we identified a research gap in applying RAG approaches in this research domain by unifying RKGs and LLMs. Therefore, we designed and implemented a training-free and schema-agnostic retrieval approach, called HubLink, to cope with the complex description and metadata types of research artifacts that are subject to change (Kaplan et al. [134]). We evaluated HubLink against three state-of-the-art baselines assessing retrieval and answer generation quality based on the extrinsic evaluation strategy for the research goal *purpose* of our evaluation approach (C1). All selected baseline approaches were published in top-tier Association for Computational Linguistics (ACL) venues between 2023 and 2025, thereby ensuring methodological rigor and validation by the research community. As knowledge base, we use the populated ORKG with our classification and data schema for SWA research (C3) by constructing four graph variants (C5). HubLink outperformed all baselines, except for *Faithfulness* in the final answer generation. The evaluation results for retrieval demonstrate that HubLink considerably improves scholarly information retrieval from RKGs, particularly through its high recall and the schema-agnostic, multi-hop retrieval capabilities. It effectively supports both simple and moderately complex queries while remaining efficient in runtime and resource usage. However, limitations in precision and ranking, especially for complex logical queries, highlight the need for downstream filtering or reranking to enhance result quality. HubLink demonstrates improved answer generation over baseline methods, particularly in capturing user intent and adhering to query instructions. However, its ability to consistently translate retrieved facts into concise and accurate answers remains limited. Lower recall, precision, and faithfulness scores indicate that not all relevant information is effectively integrated, and answers tend to be verbose. Future work should focus on refining the generation strategy to enhance conciseness and relevance (see Section 15.3). In summary, we provided a retrieval framework for benchmarking and a novel schema-agnostic, training-free retrieval approach, supporting our research goal of promoting metadata maintenance and evolution.

15.2. Relevance and Implications for Research and Practice

This doctoral thesis proposed an approach for comprehensive and systematic knowledge representation, extraction, and access based on our conceptual and technical considerations and contributions. In the following, we discuss the relevance, implications, and resulting benefits of our contributions for research and practice regarding the defined roles and perspectives in Section 1.5 that are involved in the general RLC (Figure 1.1).

Author, planning / designing a research project (prospective) and writing a publication Based on our classification and data schemas, researchers (prospective authors) are able to systematically construct a method design. Each research step can be specified on an appropriate granularity level, i.e., based on research objects, to assemble a research project. This type of classification may better position the research work and strengthen the

communication of the research project on a statement-based level. Moreover, researchers can experience and explore the variety of SE research strategies and contributions. In advance, researchers may also consider and have improved awareness of methodological and collaboration challenges. In turn, they can adopt and apply best practices in their research work. Authors benefit from semantic knowledge descriptions by accessing structured insights into prior work, datasets, and tools. RKGs enable queryable knowledge about research gaps, methodological patterns, and evaluation strategies, supporting more informed and innovative research design.

Author, reflecting on a (own) research project (retrospective) Based on our classification and data schemas, authors are able to evaluate research work in more detail and can analyze each research step and claimed contribution. They also preserve the possibility to scrutinize their own reporting style and think about different scenarios for their research steps. Shortcomings like inconsistencies between research objects and outcomes of their work can be identified and improved upon. Retrospective authors can semantically annotate their contributions (in the post-publication phase), linking findings to reusable artifacts and contextualizing results within broader knowledge structures. This enhances transparency, reproducibility, and the discoverability of their work.

Reviewer, assessing a research project Based on our classification and data schemas, reviewers can systematically analyze a publication by its research objects, contributions, and the corresponding research process. In advance, authors and reviewers should classify a publication with our schema to achieve a shared understanding for embracing diversity of research strategies. This could enforce consistency in assessing publications, promote shared agreement by building consensus among reviewers, and, thus, create transparency in the review process. Moreover, challenges that authors face are more present for the reviewers and, thus, are taken more into account. Similarly to a reflecting author, reviewers can identify shortcomings in the method design. Moreover, reviewers gain access to semantically enriched metadata and provenance information, allowing for more rigorous and efficient evaluation of methodological soundness, artifact availability, and alignment with community standards.

Evidence-based Researchers, delving into SE research Based on the current state of our schemas and the SLR results, researchers can identify research gaps, which can lead to future meta-research in SE w.r.t. the five main topic areas by Ioannidis et al. [112]. Thus, evidence-based researchers can enhance or further refine our classification schemas. They can perform systematic reviews and meta-analyses more effectively by querying structured representations of empirical evidence, evaluation methods, and validity threats across studies. RKGs, especially ORKG, support comparison of findings and sustainable literature reviews according to Santos et al. [252].

Steering Committee and Program Chairs, defining settings for a venue while evolving a research community Based on our classification and data schemas, this role benefits from semantic analytics to identify emerging research trends, underrepresented topics, and

collaboration networks. RKGs enable data-driven decisions in shaping conference themes, tracks, and reviewer assignments, fostering a coherent and inclusive research community.

Editors (especially, editors-in-chief as head of an editorial board), taking responsibility for the definition and implementation of the scientific strategy of a journal Based on our classification and data schemas, editors-in-chief can define and monitor the journal's scientific strategy by leveraging semantic descriptions of submissions, citation networks, and artifact reuse. This supports strategic curation, impact assessment, and alignment with Open Science principles.

Replicator, repeating a research project Based on our classification and data schemas, researchers obtain an overview of the varieties of replication studies and, thus, can position, design, and classify their work accordingly. Replicators can locate and reuse semantically described datasets, software, and experimental protocols. RKGs facilitate the identification of replication-relevant artifacts and contextual dependencies, reducing ambiguity, and enhancing reproducibility.

Practitioner, consulting and decision-making Based on our classification and data schemas, practitioners may better assess the relevance of research works. They may also classify publications with different reporting styles more efficiently. This allows them to compare research on the same feature criteria in a unified way (i.e., facilitating the decision-making process). Moreover, they can regard challenges in collaboration with researchers. In addition, they benefit from structured access to validated knowledge, enabling evidence-informed decisions. Semantic descriptions allow for filtering research outputs by domain relevance, maturity level, and applicability, bridging the gap between research and practice.

Knowledge Engineer and Developer, organizing and searching for relevant research Based on our classification and data schemas, knowledge engineers receive a structuring tool for supporting bibliographic concerns and indexing based on different (expert) needs. For instance, researchers might be able to search for publications employing a specific evaluation approach for a dedicated contribution. This allows them to query knowledge about contributions, research questions, or evaluation methods, instead of relying on titles, keyword-based searches, specific domains, or human expertise about related and relevant work. These roles are central to implementing and maintaining the semantic infrastructure. They design and select crucial parts of a research knowledge description approach, integrate heterogeneous data sources, and develop RAG systems that enable intelligent querying and reasoning over scholarly content in a schema-agnostic way.

A knowledge-centered approach, operationalized through domain-specific metadata specifications, semantic descriptions, and KGs, has the potential to transform scholarly communication into a machine-interpretable, interoperable, and reusable ecosystem. This paradigm enables researchers, practitioners, and other roles and stakeholders to engage with research artifacts not merely as static documents, but as dynamic, linked knowledge

entities. Such transformation enhances transparency, fosters collaboration across roles and disciplines, and accelerates scientific progress.

In this dissertation, we contribute to the structured description of research knowledge in SE, with a particular emphasis on SWA research. Our approach captures and evaluates the evolution of research artifacts over time, serving as a blueprint for constructing a populated research knowledge base (in a semi-automated way using our classification framework). Building upon this foundation, we developed a framework that supports the implementation of our schema-agnostic retrieval approach, HubLink, and enables benchmarking against state-of-the-art retrieval approaches. This integrated framework demonstrates the potential of combining semantic knowledge representation with advanced retrieval techniques to support sustainable and scalable research knowledge access for the aforementioned roles in SE research and beyond.

15.3. Outlook

Beyond the scope of this doctoral thesis and based on the main contributions w.r.t. the representation of scientific knowledge for SE research artifacts (e.g., papers, data, and software) to enable research applications such as knowledge retrieval and access, there is potential for future work, which is presented in the following for each main subcontribution in Part II and Part III.

Evaluation Approach for Classification Schemas Our evaluation approach for classification and data schemas started with a proposal-for-solution, supported by an application case example (cf. Kaplan et al. [136]). During this doctoral thesis, we further elaborated this approach by adding the evaluation strategies dimension, new properties and new metrics (e.g., combined metrics for assessing performance and carbon emission in trade-off considerations in Kaplan et al. [132]). For every GQM fragment in this evaluation approach, we could show its feasibility with the help of real-world case examples in SE research. Future work can involve the development of detailed guidelines for addressing various types of threats to validity for each evaluation strategy, including recommendations for their identification, discussion by researchers (authors), and corresponding mitigation efforts. Additionally, further research is required to ensure the completeness and robustness of all components within the approach over time. This includes the specification of research goals, the set of evaluated properties, template research and validation questions, associated metrics, and evaluation strategies. This can be done, e.g., via further literature studies to systematically map different concepts to our approach and via domain-expert interviews to ensure the maintenance of our approach. Another aspect for future work would include the investigation of the transferability and adaptability of our approach to other knowledge description approaches, such as meta-models or reference models. Ultimately, setting our evaluation approach as a general research approach and guideline, and achieving broader community acceptance for standardization would require, among other things, active work in research initiatives and further comprehensive knowledge dissemination through an extended and updated publication version of our approach.

Tool-supported Documentation Process for Literature Studies Our proposed guideline introduces a tool-supported documentation process for conducting systematic literature studies, with a particular focus on managing large-scale reviewer collaboration during the data extraction phase. This approach leverages the capabilities of modern (decentralized) version control systems, such as branching, versioning, Git-Wiki documentation, and issue tracking, to ensure traceability, transparency, and reproducibility throughout the review process. Additionally, it incorporates machine-readable formats enabled by the SLR Toolkit, including taxonomy management and support for diverse metadata catalogs. This guideline is available as a blueprint within our GitLab organization[4] in the context of our proposal of a classification and data schema for SWA research. To increase the visibility and adoption of our approach, future work may involve publishing a generalized description of the guided process in relevant venues and contributing to the refinement of methodological standards for secondary studies[5] developed by the ACM SIGSOFT Paper and Peer Review Quality Initiative (cf. Section 3.1.1). Looking ahead, we anticipate that sustainable literature reviews will become a standard practice, supported by stable and interoperable metadata formats. In particular, the adoption of semantic web technologies, such as RDF-based representations in RKGs, may emerge as a state-of-practice. Nevertheless, metadata maintenance and evolution will remain essential, underscoring the importance of lightweight, adaptable documentation approaches like ours that are easy to set up and integrate into existing research workflows.

Classification and Data Schema for Software Architecture Research To unlock the full potential of digitalization in research and to promote sustainable research data management, we need a robust classification and data schema for describing the metadata of research artifacts. To construct such an approach, we started with a general concept by describing papers and their supplementary materials based on their core *Findings* and the underlying *Validity* and *Evidence*. We instantiated this approach with a domain-specific use case in SE research, focusing on SWA research. So far, we demonstrated the feasibility of our approach in a literature study by describing current practices in evaluating SWA research objects and partly by our evaluation approach for classification schemas. Future work would include investigating the perception of our proposed schema, e.g., via domain-expert interviews or community surveys, to gain more insights. Furthermore, we envision collaborative efforts with the SWA research community to organize paper annotation tasks in conjunction with artifact submissions at relevant conferences and workshops. Such initiatives would align with existing workflows in other SE research subfields, such as the annotation task introduced in the Open Science Track at the International Working Conference on Requirements Engineering (REFSQ)[6] in 2024. Notably, for the 2026 edition[7], the annotation challenge refers to parts of our classification schema, e.g., considering the *Evaluation Method* or adapting the *Paper Class*. These community-driven activities will

[4] `https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice` [Last accessed on 2025-08-21]

[5] `https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=SystematicReviews` [Last accessed on 2025-08-21]

[6] `https://2024.refsq.org/track/refsq-2024-open-science` [Last accessed on 2025-09-06]

[7] `https://2026.refsq.org/track/refsq-2026-open-science#Open-Science-Competition` [Last accessed on 2025-09-06]

not only support the maturation of our proposed schema but also facilitate the systematic collection of metadata for reuse in future research and tooling applications. Ultimately, these efforts aim to foster a culture of transparent, reusable, and semantically rich research documentation, contributing to the broader goals of Open Science and sustainable, knowledge-centered scholarly communication.

Unified and Maintenance Approach for Knowledge Description The classification and data schema for SWA research (or in general) are subject to evolution over time, primarily driven by shifts in research practices within the SE community. These changes may be captured descriptively, reflecting current practices, or normatively, with a forward-looking intent to guide future developments. To systematically address such scenarios, we have developed a general research process grounded in our evaluation approach (C1), enabling the tracking of versions and variants of classification and metadata schemas for research artifacts. Future work will focus on maintaining and curating our schema repository to ensure that derived and enhanced versions remain up-to-date and aligned with emerging community standards. Additionally, further investigation is needed to understand how the general schema for SE influences and informs the more specialized schema for SWA. This includes exploring opportunities to expand the SWA schema to support more granular and semantically rich metadata descriptions. These efforts should be coordinated with ongoing and future initiatives aimed at refining the initial SWA schema, ensuring consistency and interoperability across SWA subdomains. By fostering a flexible yet structured approach to schema changes, we aim to support sustainable metadata management and facilitate reuse in both empirical research and tool development.

ORKG Template Specification We transferred our classification and data schema for SWA to the concept of ORKG templates. This allowed us to explore the capabilities of ORKG for structured metadata representation and to construct a knowledge base supporting the evaluation of our retrieval approach (C7). As part of this effort, we initiated an ORKG Observatory[8], which serves as a curated environment for monitoring and analyzing metadata contributions. Future work will focus on incorporating insights gained from ongoing schema development and community feedback, as previously presented. As our understanding of metadata requirements evolves, particularly through domain-specific studies and expert engagement, we aim to refine and extend the schema to better reflect the needs of the research community. The current ORKG template follows a hierarchical structure, which is widely adopted and well-understood. However, related work in engineering sciences has specified cyclic descriptions within ORKG templates, suggesting that certain research processes may be better represented through non-linear structures. In this context, we plan to investigate the role of cyclic descriptions in SE and SWA research, particularly in modeling research steps (cf. Section 2.1.2). This exploration may lead to more expressive and accurate representations of scholarly workflows, enhancing the semantic richness and utility of the ORKG-based metadata framework.

[8] `https://orkg.org/observatories/Software_Architecture_Research` [Last accessed on 2025-08-21]

Best Practices for AI-based Studies The digital change in research is crucial to cope with the rapidly growing amount of research knowledge and insights. This change is mainly driven by the specification of metadata types and AI techniques, such as the emergence of LLMs and semantic web technologies. Thus, we can observe a need for community guidelines for empirical studies in SE involving LLMs, as outlined by Wagner et al. [298]. We see the potential for complementary integration into the existing guideline draft[9] based on our work in Kaplan et al. [132, 134]. In our evaluation approach (C1), this would affect the machine-related evaluation strategy for the research goal *applicability* and the extrinsic evaluation strategy for the research goal *purpose*. By combining these works, future LLM studies can benefit from the following aspects: The integration of energy-aware metrics into the LLM guidelines enables a holistic evaluation approach that accounts for both functional performance and environmental impact. This multidimensional perspective is essential for aligning LLM development with broader goals of sustainability and responsible AI deployment (Green AI). To further support sustainable AI practices, prompt engineering guidelines can be extended to include constraints on prompt complexity and inference depth. These constraints aim to reduce unnecessary computational overhead, thereby improving energy efficiency without compromising model utility. Transparent decision-making is another critical dimension of responsible LLM usage. By incorporating sustainability trade-offs into the documentation and evaluation process, model behavior becomes more traceable and reproducible. This transparency ensures that design choices are aligned with research objectives and ethical considerations. The proposed and combined guidelines should be adaptable across domains, including scholarly communication, software engineering, and public services. In these contexts, both performance and sustainability are paramount, and a unified approach can facilitate consistent and responsible model deployment. Importantly, the inclusion of RAG as a new study type within the LLM guidelines would enhance their applicability. This integration fosters the development of LLM systems that are not only trustworthy and adaptive but also resource-efficient and contextually grounded.

Automated Classification Framework Our automated classification framework encompasses multiple features and aspects of variations w.r.t. the experimental setup to derive suitable LLM-based classifiers. We support multi-label classification problems, being able to support complex metadata types of research artifacts. As our framework is a powerful NLP tool and already characterized by a high level of maturity, we recommend the integration of two further agents that we already implemented in preliminary work, based on our obtained evaluation results from our case example. First, we aim to support full-text integration for the datasets. Therefore, we recommend an agent that integrates LLMLingua[10] for text compression. In this way, we can deal with poor information content in abstracts and support lower inference costs and processing time during the classification task. Second, we aim to cope with unstructured outputs of LLM results by integrating Instructor[11] as an additional agent in our framework while fostering the robustness of

[9] `https://llm-guidelines.org/` [Last accessed on 2025-08-21]
[10] `https://github.com/microsoft/LLMLingua` [Last accessed on 2025-08-21]
[11] `https://python.useinstructor.com/why/`[Last accessed on 2025-08-21]

evaluating and calculating the classification results. To further highlight the feasibility, usefulness, and generalizability of our framework and its features, we aim to apply it to other classification schemas provided in the SE literature (e.g., the classification and data schema for software architecture optimization methods by Aleti et al. [5]) and beyond.

Retrieval Framework and Approach For research knowledge access, we designed and implemented a retrieval framework and a retrieval approach, called HubLink. The retrieval approach works in a schema-agnostic and training-free manner, supporting the overall objective of this doctoral thesis to enable the maintenance and evolution of research knowledge description approaches. Our evaluation of HubLink revealed limitations in the alignment between retrieved contexts and generated answers. Future work should explore improved alignment strategies, such as advanced prompt engineering or alternative synthesis methods beyond single-prompt generation. As presented in Section 10.3, our crafted KGQA dataset used for our experiments operates at a label-based granularity. Applying HubLink to datasets with different (text) granularity levels could validate its generalizability and assess the utility of its linking feature, which was underutilized in our current experimental setup. We also highlighted HubLink's difficulty in ranking relevant contexts above irrelevant ones. Therefore, enhancing retrieval relevance through refined prompts or post-retrieval re-ranking mechanisms is a promising direction. Extending the hub representations in the RKG with additional information, such as summaries or precomputed numerical aggregations, may enrich retrieval and synthesis, potentially improving answer quality. While HubLink could handle basic temporal constraints, its performance on complex numerical reasoning remains unexplored. Future work should evaluate its capabilities on queries involving intricate temporal or quantitative filters. Beyond graph-based retrieval, HubLink could be adapted for document retrieval scenarios, using the KG as a supporting structure. Preliminary work has laid the foundation for this, including baseline implementations of LightRAG[12] and GraphRAG[13]. However, a full evaluation remains pending. The evaluation of our retrieval approach, HubLink, was conducted on the ORKG as knowledge base. Future work should assess its generalizability across diverse RDF-based RKGs such as the dblp Knowledge Graph[14] (dblp KG) using our retrieval framework. Finally, to improve usability, future work can integrate the current CLI-based system into a graphical user interface (GUI) like Open WebUI[15]. This will expose semantic search and reasoning capabilities to non-technical users, with planned features including modular APIs and dynamic KG visualizations to support intuitive exploration and query refinement.

[12] `https://github.com/HKUDS/LightRAG` [Last accessed on 2025-08-21]
[13] `https://github.com/microsoft/graphrag` [Last accessed on 2025-08-21]
[14] `https://blog.dblp.org/tag/knowledge-graph/` [Last accessed on 2025-08-20]
[15] `https://docs.openwebui.com/` [Last accessed on 2025-08-20]

Bibliography

[1] Marcel R. Ackermann, Hannah Bast, Benedikt Maria Beckermann, Johannes Kalmbach, Patrick Neises, and Stefan Ollinger. "The dblp Knowledge Graph and SPARQL Endpoint". In: *Transactions on Graph Data and Knowledge* 2.2 (2024), 3:1–3:23. ISSN: 2942-7517. DOI: 10.4230/TGDK.2.2.3. URL: https://drops.dagstuhl.de/entities/document/10.4230/TGDK.2.2.3.

[2] Russell L. Ackoff. "From data to wisdom". In: *Journal of Applied Systems Analysis* 16.1 (1989), pp. 3–9. URL: https://faculty.ung.edu/kmelton/Documents/DataWisdom.pdf.

[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. Ed. by Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis. ACM, 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701. URL: https://doi.org/10.1145/3292500.3330701.

[4] Rand Alchokr, Jacob Krüger, Yusra Shakeel, Gunter Saake, and Thomas Leich. "Peer-Reviewing and Submission Dynamics Around Top Software-Engineering Venues: A Juniors' Perspective". In: *EASE 2022: The International Conference on Evaluation and Assessment in Software Engineering 2022, Gothenburg, Sweden, June 13 - 15, 2022*. Ed. by Miroslaw Staron, Christian Berger, Jocelyn Simmonds, and Rafael Prikladnicki. ACM, 2022, pp. 60–69. DOI: 10.1145/3530019.3530026. URL: https://doi.org/10.1145/3530019.3530026.

[5] Aldeida Aleti, Barbora Buhnova, Lars Grunske, Anne Koziolek, and Indika Meedeniya. "Software Architecture Optimization Methods: A Systematic Literature Review". In: *IEEE Transactions on Software Engineering* 39.5 (2013), pp. 658–683. DOI: 10.1109/TSE.2012.64. URL: https://doi.org/10.1109/TSE.2012.64.

[6] Ashkan Alinejad, Krtin Kumar, and Ali Vahdat. "Evaluating the Retrieval Component in LLM-Based Question Answering Systems". In: *CoRR* abs/2406.06458 (2024). DOI: 10.48550/ARXIV.2406.06458. URL: https://doi.org/10.48550/arXiv.2406.06458.

[7] João Franscisco Almeida and António Rito Silva. "Monolith Migration Complexity Tuning Through the Application of Microservices Patterns". In: *Software Architecture - 14th European Conference, ECSA 2020, L'Aquila, Italy, September 14-18, 2020, Proceedings*. Ed. by Anton Jansen, Ivano Malavolta, Henry Muccini, Ipek Ozkaya, and Olaf Zimmermann. Vol. 12292. Lecture Notes in Computer Science.

Springer, 2020, pp. 39–54. DOI: 10.1007/978-3-030-58923-3_3. URL: https://doi.org/10.1007/978-3-030-58923-3_3.

[8] Apostolos Ampatzoglou, Stamatia Bibi, Paris Avgeriou, Marijn Verbeek, and Alexander Chatzigeorgiou. "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies". In: *Information and Software Technology* 106 (2019), pp. 201–230. DOI: 10.1016/J.INFSOF.2018.10.006. URL: https://doi.org/10.1016/j.infsof.2018.10.006.

[9] Sofia Ananieva, Sandra Greiner, Timo Kehrer, Jacob Krüger, Thomas Kühn, Lukas Linsbauer, Sten Grüner, Anne Koziolek, Henrik Lönn, S. Ramesh, and Ralf H. Reussner. "A conceptual model for unifying variability in space and time: Rationale, validation, and illustrative applications". In: *Empirical Software Engineering* 27.5 (2022), p. 101. DOI: 10.1007/S10664-021-10097-Z. URL: https://doi.org/10.1007/s10664-021-10097-z.

[10] Sofia Ananieva, Sandra Greiner, Thomas Kühn, Jacob Krüger, Lukas Linsbauer, Sten Grüner, Timo Kehrer, Heiko Klare, Anne Koziolek, Henrik Lönn, Sebastian Krieter, Christoph Seidl, S. Ramesh, Ralf H. Reussner, and Bernhard Westfechtel. "A conceptual model for unifying variability in space and time". In: *SPLC '20: 24th ACM International Systems and Software Product Line Conference, Montreal, Quebec, Canada, October 19-23, 2020, Volume A*. Ed. by Roberto Erick Lopez-Herrejon. ACM, 2020, 15:1–15:12. DOI: 10.1145/3382025.3414955. URL: https://doi.org/10.1145/3382025.3414955.

[11] Matej Artac, Tadej Borovsak, Elisabetta Di Nitto, Michele Guerriero, Diego Perez-Palacin, and Damian Andrew Tamburri. "Infrastructure-as-Code for Data-Intensive Architectures: A Model-Driven Development Approach". In: *IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018*. IEEE Computer Society, 2018, pp. 156–165. DOI: 10.1109/ICSA.2018.00025. URL: https://doi.org/10.1109/ICSA.2018.00025.

[12] Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G. Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. "The SciQA Scientific Question Answering Benchmark for Scholarly Knowledge". In: *Scientific Reports* 13.1 (2023), p. 7240. URL: https://doi.org/10.1038/s41598-023-33607-z.

[13] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria-Esther Vidal. "Towards a Knowledge Graph for Science". In: *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018*. Ed. by Rajendra Akerkar, Mirjana Ivanovic, Sang-Wook Kim, Yannis Manolopoulos, Riccardo Rosati, Milos Savic, Costin Badica, and Milos Radovanovic. ACM, 2018, 1:1–1:6. DOI: 10.1145/3227609.3227689. URL: https://doi.org/10.1145/3227609.3227689.

[14] Sören Auer, Allard Oelen, Muhammad Haris, Markus Stocker, Jennifer D'Souza, Kheir Eddine Farfar, Lars Vogt, Manuel Prinz, Vitalis Wiens, and Mohamad Yaser Jaradeh. "Improving access to scientific literature with knowledge graphs". In: *Bibliothek Forschung und Praxis* 44.3 (2020), pp. 516–529.

[15] Alberto Avritzer, Vincenzo Ferme, Andrea Janes, Barbara Russo, Henning Schulz, and André van Hoorn. "A Quantitative Approach for the Assessment of Microservice Architecture Deployment Alternatives by Automated Performance Testing". In: *Software Architecture - 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24-28, 2018, Proceedings*. Ed. by Carlos E. Cuesta, David Garlan, and Jennifer Pérez. Vol. 11048. Lecture Notes in Computer Science. Springer, 2018, pp. 159–174. DOI: 10.1007/978-3-030-00761-4_11. URL: https://doi.org/10.1007/978-3-030-00761-4_11.

[16] Athanasios Bachoumis, Costas Mylonas, Konstantinos Plakas, Michael K. Birbas, and Alexios N. Birbas. "Data-Driven Analytics for Reliability in the Buildings-to-Grid Integrated System Framework: A Systematic Text-Mining-Assisted Literature Review and Trend Analysis". In: *IEEE Access* 11 (2023), pp. 130763–130787. DOI: 10.1109/ACCESS.2023.3335191. URL: https://doi.org/10.1109/ACCESS.2023.3335191.

[17] Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. "Direct Fact Retrieval from Knowledge Graphs without Entity Linking". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 10038–10055. DOI: 10.18653/v1/2023.acl-long.558. URL: https://aclanthology.org/2023.acl-long.558/.

[18] Debayan Banerjee, Sushil Awale, Ricardo Usbeck, and Chris Biemann. "DBLP-QuAD: A Question Answering Dataset over the DBLP Scholarly Knowledge Graph". In: *Proceedings of the 13th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 45th European Conference on Information Retrieval (ECIR 2023), Dublin, Ireland, April 2nd, 2023*. Ed. by Ingo Frommholz, Philipp Mayr, Guillaume Cabanac, Suzan Verberne, and Jordan Brennan. Vol. 3617. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 37–51. URL: https://ceur-ws.org/Vol-3617/paper-05.pdf.

[19] Victor R. Basili, Gianluigi Caldiera, and Dieter H. Rombach. "The goal question metric approach". In: *Encyclopedia of Software Engineering* (1994), pp. 528–532.

[20] Victor R. Basili and David M. Weiss. "A Methodology for Collecting Valid Software Engineering Data". In: *IEEE Transactions on Software Engineering* 10.6 (1984), pp. 728–738. DOI: 10.1109/TSE.1984.5010301. URL: https://doi.org/10.1109/TSE.1984.5010301.

[21] Sussy Bayona-Oré, Jose A Calvo-Manzano, Gonzalo Cuevas, and Tomas San-Feliu. "Critical success factors taxonomy for software process deployment". In: *Software Quality Journal* 22.1 (2014), pp. 21–48. DOI: 10.1007/S11219-012-9190-Y. URL: https://doi.org/10.1007/s11219-012-9190-y.

[22] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Martin Mahaux, Birgit Penzenstadler, Guillermo Rodriguez-Navas, Camille Salinesi, Norbert Seyff, Colin C. Venters, Coral Calero, Sedef Akinli Koçak, and Stefanie Betz.

"The Karlskrona manifesto for sustainability design". In: *CoRR* abs/1410.6968 (2014). arXiv: 1410.6968. URL: http://arxiv.org/abs/1410.6968.

[23] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. "Sustainability Design and Software: The Karlskrona Manifesto". In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 2. 2015, pp. 467–476. DOI: 10.1109/ICSE.2015.179.

[24] Denise Bedford. "Evaluating classification schema and classification decisions". In: *Bulletin of the American Society for Information Science and Technology* 39.2 (2013), pp. 13–21. DOI: 10.1002/bult.2013.1720390206.

[25] Iz Beltagy, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 3613–3618. DOI: 10.18653/V1/D19-1371. URL: https://doi.org/10.18653/v1/D19-1371.

[26] David Benavides, Chico Sundermann, Kevin Feichtinger, José A. Galindo, Rick Rabiser, and Thomas Thüm. "UVL: Feature modelling with the Universal Variability Language". In: *Journal of Systems and Software* 225 (2025), p. 112326. DOI: 10.1016/J.JSS.2024.112326. URL: https://doi.org/10.1016/j.jss.2024.112326.

[27] Edward M. Bennett, Renee Alpert, and AC Goldstein. "Communications through limited-response questioning". In: *Public Opinion Quarterly* 18.3 (1954), pp. 303–308.

[28] Tim Berners-Lee and James Hendler. "Publishing on the semantic web". In: *Nature* 410.6832 (2001), pp. 1023–1024. DOI: 10.1038/35074206. URL: https://doi.org/10.1038/35074206.

[29] Antonia Bertolino, Antonello Calabrò, Francesca Lonetti, Eda Marchetti, and Breno Miranda. "A Categorization Scheme for Software Engineering Conference Papers and its Application". In: *Journal of Systems and Software* 137 (2018), pp. 114–129. DOI: 10.1016/j.jss.2017.11.048. URL: https://doi.org/10.1016/j.jss.2017.11.048.

[30] Manoj Bhat, Klym Shumaiev, Andreas Biesdorf, Uwe Hohenstein, and Florian Matthes. "Automatic Extraction of Design Decisions from Issue Management Systems: A Machine Learning Based Approach". In: *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*. Ed. by Antónia Lopes and Rogério de Lemos. Vol. 10475. Lecture Notes in Computer Science. Springer, 2017, pp. 138–154. DOI: 10.1007/978-3-319-65831-5_10. URL: https://doi.org/10.1007/978-3-319-65831-5_10.

[31] Rainer Böhme and Ralf H. Reussner. "Validation of Predictions with Measurements". In: *Dependability Metrics: Advanced Lectures [result from a Dagstuhl seminar, October 30 - November 1, 2005]*. Ed. by Irene Eusgeld, Felix C. Freiling, and Ralf H. Reussner. Vol. 4909. Lecture Notes in Computer Science. Springer, 2005, pp. 14–18. DOI: 10.1007/978-3-540-68947-8_3. URL: https://doi.org/10.1007/978-3-540-68947-8_3.

[32] Ronald F. Boisvert. "Incentivizing reproducibility". In: *Communications of the ACM* 59.10 (2016), p. 5. DOI: 10.1145/2994031. URL: https://doi.org/10.1145/2994031.

[33] Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W. Bruce Croft, and Mark Sanderson. "A Non-Factoid Question-Answering Taxonomy". In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '22. Madrid, Spain: Association for Computing Machinery, 2022, pp. 1196–1207. ISBN: 9781450387323. DOI: 10.1145/3477495.3531926.

[34] Semen A. Budennyy, Vladimir D. Lazarev, Nikita Zakharenko, Alexey N. Korovin, Olga Plosskaya, Denis Dimitrov, Vladimir Arkhipkin, I. V. Pavlov, Ivan V. Oseledets, Ivan Barsola, Ilya Egorov, Aleksandra Kosterina, and Leonid Zhukov. "Eco2AI: carbon emissions tracking of machine learning models as the first step towards sustainable AI". In: *Doklady Mathematics* 106 (2022). DOI: 10.1134/S1064562422060230. URL: https://doi.org/10.1134/S1064562422060230.

[35] David Budgen, Barbara A. Kitchenham, Stuart M. Charters, Mark Turner, Pearl Brereton, and Stephen G. Linkman. "Presenting software engineering results using structured abstracts: a randomised experiment". In: *Empirical Software Engineering* 13.4 (2008), pp. 435–468. DOI: 10.1007/S10664-008-9075-7. URL: https://doi.org/10.1007/s10664-008-9075-7.

[36] Jungwon Byun and Andreas Stuhlmüller. *Elicit: Language models as research tools*. Paris: OECD Publishing, 2023. DOI: https://doi.org/10.1787/174aee8f-en.

[37] Marçal Mora Cantallops, Salvador Sánchez-Alonso, Elena García-Barriocanal, and Miguel-Ángel Sicilia. "Traceability for Trustworthy AI: A Review of Models and Tools". In: *Big Data and Cognitive Computing* 5.2 (2021), p. 20. DOI: 10.3390/BDCC5020020. URL: https://doi.org/10.3390/bdcc5020020.

[38] Paulo Vinicius Cardoso, Rhauani Weber Aita Fazul, and Patrícia Pitthan Barcelos. "Employment of Optimal Approximations on Apache Hadoop Checkpoint Technique for Performance Improvements". In: *2020 IEEE International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, March 16-20, 2020*. IEEE, 2020, pp. 1–10. DOI: 10.1109/ICSA47634.2020.00009. URL: https://doi.org/10.1109/ICSA47634.2020.00009.

[39] Jeffrey C. Carver, Edgar E. Hassler, Elis Hernandes, and Nicholas A. Kraft. "Identifying Barriers to the Systematic Literature Review Process". In: *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, Maryland, USA, October 10-11, 2013*. IEEE Computer Society, 2013, pp. 203–212. DOI: 10.1109/ESEM.2013.28. URL: https://doi.org/10.1109/ESEM.2013.28.

[40] Miquel Centelles and Núria Ferran-Ferrer. "Assessing knowledge organization systems from a gender perspective: Wikipedia taxonomy and Wikidata ontologies". In: *Journal of Documentation* 80.7 (2024), pp. 124–147. DOI: 10.1108/JD-11-2023-0230. URL: https://www.emerald.com/insight/content/doi/10.1108/jd-11-2023-0230/full/html.

[41] Fatma Chebbi, Dominik Fuchß, and Angelika Kaplan. *Visulite (VISUalization of LITErature)*. 2023. URL: https://softwarearchitectureresearch.github.io/visulite/.

[42] Fatma Chebbi, Dominik Fuchß, and Angelika Kaplan. *Visulite (VISUalization of LITErature) Docs*. 2023. URL: https://github.com/SoftwareArchitectureResearch/visulite.

[43] K. R. Chowdhary. "Natural Language Processing". In: *Fundamentals of Artificial Intelligence* (2020), pp. 603–649. DOI: 10.1007/978-81-322-3972-7_19. URL: https://doi.org/10.1007/978-81-322-3972-7_19.

[44] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. "SPECTER: Document-level Representation Learning using Citation-informed Transformers". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, 2020, pp. 2270–2282. DOI: 10.18653/V1/2020.ACL-MAIN.207. URL: https://doi.org/10.18653/v1/2020.acl-main.207.

[45] Jacob Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[46] Thomas Cook. *Quasi-experimentation: design & analysis issues for field settings*. Boston: Houghton Mifflin, 1979. ISBN: 9780395307908.

[47] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, Marion Coutarel, Boris Feld, Jérémy Lecourt, Liam Connell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. *mlco2/codecarbon: v2.4.1*. Version v2.4.1. May 2024. DOI: 10.5281/zenodo.11171501. URL: https://doi.org/10.5281/zenodo.11171501.

[48] Margarita Cruz, Beatriz Bernárdez, Amador Durán, José Angel Galindo, and Antonio Ruiz-Cortés. "Replication of Studies in Empirical Software Engineering: A Systematic Mapping Study, From 2013 to 2018". In: *IEEE Access* 8 (2020), pp. 26773–26791. DOI: 10.1109/ACCESS.2019.2952191. URL: https://doi.org/10.1109/ACCESS.2019.2952191.

[49] Daniela S. Cruzes and Tore Dybå. "Synthesizing evidence in software engineering research". In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, Sept. 2010. DOI: 10.1145/1852786.1852788.

[50] Susmita Das, Amara Tariq, Thiago Santos, Sai Sandeep Kantareddy, and Imon Banerjee. "Recurrent neural networks (RNNs): architectures, training tricks, and introduction to influential research". In: *Machine learning for Brain disorders* (2023), pp. 117–138. DOI: 10.1007/978-1-0716-3195-9_4. URL: https://doi.org/10.1007/978-1-0716-3195-9_4.

[51] Charles Despres and Daniele Chauvel. "Knowledge management (s)". In: *Journal of Knowledge Management* 3.2 (1999), pp. 110–123.

[52] Danilo Dessì, Francesco Osborne, Davide Buscaldi, Diego Reforgiato Recupero, and Enrico Motta. "CS-KG 2.0: A Large-scale Knowledge Graph of Computer Science". In: *Scientific Data* 12.1 (2025), p. 964. DOI: 10.1038/s41597-025-05200-8. URL: https://doi.org/10.1038/s41597-025-05200-8.

[53] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. "CS-KG: A Large-Scale Knowledge Graph of Research Entities and Claims in Computer Science". In: *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*. Ed. by Ulrike Sattler, Aidan Hogan, C. Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato. Vol. 13489. Lecture Notes in Computer Science. Springer, 2022, pp. 678–696. DOI: 10.1007/978-3-031-19433-7_39. URL: https://doi.org/10.1007/978-3-031-19433-7_39.

[54] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

[55] J. T. Dillon. "The Classification of Research Questions". In: *Review of Educational Research* 54.3 (1984), pp. 327–361. DOI: 10.3102/00346543054003327.

[56] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. "The Faiss library". In: *CoRR* abs/2401.08281 (2024). DOI: 10.48550/ARXIV.2401.08281. URL: https://doi.org/10.48550/arXiv.2401.08281.

[57] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. "LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia". In: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*. Ed. by Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon. Springer International Publishing, 2019, pp. 69–78. ISBN: 978-3-030-30796-7. DOI: 10.1007/978-3-030-30796-7_5. URL: https://doi.org/10.1007/978-3-030-30796-7_5.

[58] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. "Selecting Empirical Methods for Software Engineering Research". In: *Guide to Advanced Empirical Software Engineering*. Springer London, 2008, pp. 285–311. DOI: 10.1007/978-1-84800-044-5_11.

[59] Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. "RAGAs: Automated Evaluation of Retrieval Augmented Generation". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations, St. Julians, Malta, March 17-22, 2024*. Ed. by Nikolaos Aletras and Orphée De Clercq. Association for Computational Linguistics, 2024, pp. 150–158. URL: https://aclanthology.org/2024.eacl-demo.16.

[60] Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Parteek Sharma, Fan Chen, and Lei Jiang. "LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models". In: *CoRR* (2023). DOI: 10.48550/ARXIV.2309.14393. URL: https://doi.org/10.48550/arXiv.2309.14393.

[61] Davide Falessi, Muhammad Ali Babar, Giovanni Cantone, and Philippe Kruchten. "Applying empirical software engineering to software architecture: challenges and lessons learned". In: *Empirical Software Engineering* 15.3 (2010), pp. 250–276. URL: https://doi.org/10.1007/s10664-009-9121-0.

[62] Davide Falessi, Philippe Kruchten, and Giovanni Cantone. "Issues in Applying Empirical Software Engineering to Software Architecture". In: *Software Architecture, First European Conference, ECSA 2007, Aranjuez, Spain, September 24-26, 2007, Proceedings*. Ed. by Flávio Oquendo. Vol. 4758. Lecture Notes in Computer Science. Springer, 2007, pp. 257–262. DOI: 10.1007/978-3-540-75132-8_20. URL: https://doi.org/10.1007/978-3-540-75132-8_20.

[63] Edward A. Feigenbaum. "Knowledge engineering". In: *Annals of the New York Academy of Sciences* 426.1 (1984), pp. 91–107.

[64] Robert Feldt and Ana Magazinius. "Validity Threats in Empirical Software Engineering Research - An Initial Survey". In: *Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010*. Knowledge Systems Institute Graduate School, 2010, pp. 374–379.

[65] Chao Feng, Xinyu Zhang, and Zichu Fei. "Knowledge Solver: Teaching LLMs to Search for Domain Knowledge from Knowledge Graphs". In: *CoRR* abs/2309.03118 (2023). DOI: 10.48550/ARXIV.2309.03118. URL: https://doi.org/10.48550/arXiv.2309.03118.

[66] Nicola Ferro and Diane Kelly. "SIGIR Initiative to Implement ACM Artifact Review and Badging". In: *SIGIR Forum* 52.1 (Aug. 2018), pp. 4–10. ISSN: 0163-5840. DOI: 10.1145/3274784.3274786. URL: https://doi.org/10.1145/3274784.3274786.

[67] Juarez L. M. Filho, Lincoln S. Rocha, Rossana M. C. Andrade, and Ricardo Britto. “Preventing Erosion in Exception Handling Design Using Static-Architecture Conformance Checking”. In: *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*. Ed. by Antónia Lopes and Rogério de Lemos. Vol. 10475. Lecture Notes in Computer Science. Springer, 2017, pp. 67–83. DOI: 10.1007/978-3-319-65831-5_5. URL: https://doi.org/10.1007/978-3-319-65831-5_5.

[68] Joseph L. Fleiss. “Measuring nominal scale agreement among many raters.” In: *Psychological bulletin* 76.5 (1971), p. 378.

[69] Martin H. Frické. “Data-Information-Knowledge-Wisdom (DIKW) Pyramid, Framework, Continuum”. In: *Encyclopedia of Big Data*. Ed. by Laurie A. Schintler and Connie L. McNeely. Cham: Springer International Publishing, 2022, pp. 364–367. ISBN: 978-3-319-32010-6. DOI: 10.1007/978-3-319-32010-6_331. URL: https://doi.org/10.1007/978-3-319-32010-6_331.

[70] Jonathan Furner. “Definitions of "Metadata": A Brief Survey of International Standards”. In: *Journal of the Association for Information Science and Technology* 71.6 (2020), E33–E42. DOI: 10.1002/ASI.24295. URL: https://doi.org/10.1002/asi.24295.

[71] Matthias Galster and Danny Weyns. “Empirical research in software architecture - Perceptions of the community”. In: *Journal of Systems and Software* 202 (2023), p. 111684. DOI: 10.1016/J.JSS.2023.111684. URL: https://doi.org/10.1016/j.jss.2023.111684.

[72] Matthias Galster and Danny Weyns. “Empirical Research in Software Architecture: How Far have We Come?” In: *13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, Venice, Italy, April 5-8, 2016*. IEEE Computer Society, 2016, pp. 11–20. DOI: 10.1109/WICSA.2016.10. URL: https://doi.org/10.1109/WICSA.2016.10.

[73] Matthias Galster, Danny Weyns, Antony Tang, Rick Kazman, and Mehdi Mirakhorli. “From craft to science: the road ahead for empirical software engineering research”. In: *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results, ICSE (NIER) 2018, Gothenburg, Sweden, May 27 - June 03, 2018*. Ed. by Andrea Zisman and Sven Apel. ACM, 2018, pp. 77–80. DOI: 10.1145/3183399.3183421. URL: https://doi.org/10.1145/3183399.3183421.

[74] Vahid Garousi, Michael Felderer, and Mika V. Mäntylä. “The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature”. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, EASE 2016, Limerick, Ireland, June 01 - 03, 2016*. Ed. by Sarah Beecham, Barbara A. Kitchenham, and Stephen G. MacDonell. ACM, 2016, 26:1–26:6. DOI: 10.1145/2915970.2916008. URL: https://doi.org/10.1145/2915970.2916008.

[75] Vahid Garousi and João M. Fernandes. "Highly-cited papers in software engineering: The top-100". In: *Information and Software Technology* 71 (2016), pp. 108–128. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2015.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0950584915001871.

[76] Susan Gasson. "Rigor in Grounded Theory Research: An Interpretive Perspective on Generating Theory from Qualitative Field Studies". In: *The Handbook of Information Systems Research*. IGI Global, 2004, pp. 79–102.

[77] Patric Genfer and Uwe Zdun. "Identifying Domain-Based Cyclic Dependencies in Microservice APIs Using Source Code Detectors". en. In: *Software Architecture - 15th European Conference, ECSA 2021*. Ed. by Stefan Biffl, Elena Navarro, Welf Löwe, Marjan Sirjani, Raffaela Mirandola, and Danny Weyns. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 207–222. ISBN: 978-3-030-86044-8. DOI: 10.1007/978-3-030-86044-8_15.

[78] Carlo Ghezzi. *Being a Researcher - An Informatics Perspective*. Springer, 2020. ISBN: 978-3-030-45156-1. DOI: 10.1007/978-3-030-45157-8. URL: https://doi.org/10.1007/978-3-030-45157-8.

[79] Barney G. Glaser and Anselm L. Strauss. *Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, 1967.

[80] Robert L. Glass, Iris Vessey, and Venkataraman Ramesh. "Research in software engineering: an analysis of the literature". In: *Information and Software Technology* 44.8 (2002), pp. 491–506. DOI: 10.1016/S0950-5849(02)00049-6. URL: https://doi.org/10.1016/S0950-5849(02)00049-6.

[81] Omar S. Gómez, Natalia Juristo Juzgado, and Sira Vegas. "Understanding replication of experiments in software engineering: A classification". In: *Information and Software Technology* 56.8 (2014), pp. 1033–1048. DOI: 10.1016/j.infsof.2014.04.004. URL: https://doi.org/10.1016/j.infsof.2014.04.004.

[82] Sebastian Götz. "Supporting Systematic Literature Reviews in Computer Science: The Systematic Literature Review Toolkit". In: *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '18. Copenhagen, Denmark: Association for Computing Machinery, 2018, pp. 22–26. ISBN: 9781450359658. DOI: 10.1145/3270112.3270117. URL: https://doi.org/10.1145/3270112.3270117.

[83] IBM Granite Team. *Granite 3.0 Language Models*. [Last Accessed on 2025-09-01]. Oct. 2024. URL: https://github.com/ibm-granite/granite-3.0-language-models/.

[84] Aaron Grattafiori and et al. *The Llama 3 Herd of Models*. [Last Accessed on 2025-09-01. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[85] Jeff Gray and Bernhard Rumpe. "How to Write a Successful SoSyM Submission". In: *Software & Systems Modeling* 15.4 (2016), pp. 929–931. DOI: 10.1007/s10270-016-0558-5. URL: https://doi.org/10.1007/s10270-016-0558-5.

[86] Thomas R. Gruber. “A translation approach to portable ontology specifications”. In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220. URL: https://doi.org/10.1006/knac.1993.1008.

[87] Michael Grüninger and Mark S. Fox. “The role of competency questions in enterprise engineering”. In: *Benchmarking–Theory and Practice*. Springer, 1995, pp. 22–31. URL: https://link.springer.com/content/pdf/10.1007/978-0-387-34847-6_3.

[88] Giancarlo Guizzardi, Luís Ferreira Pires, and Marten van Sinderen. “An Ontology-Based Approach for Evaluating the *Domain Appropriateness* and *Comprehensibility Appropriateness* of Modeling Languages”. In: *Model Driven Engineering Languages and Systems, 8th International Conference, MoDELS 2005, Montego Bay, Jamaica, October 2-7, 2005, Proceedings*. Ed. by Lionel C. Briand and Clay Williams. Vol. 3713. Lecture Notes in Computer Science. Springer, 2005, pp. 691–705. DOI: 10.1007/11557432_51. URL: https://doi.org/10.1007/11557432_51.

[89] Corin A. Gurr. “Effective diagrammatic communication: Syntactic, semantic and pragmatic issues”. In: *Journal of Visual Languages & Computing* 10.4 (1999), pp. 317–342. URL: https://doi.org/10.1006/jvlc.1999.0130.

[90] Sebastian Hahner, Stephan Seifermann, Robert Heinrich, and Ralf H. Reussner. “A Classification of Software-Architectural Uncertainty Regarding Confidentiality”. In: *E-Business and Telecommunications: 18th International Conference, ICETE 2021, Virtual Event, July 6–9, 2021, Revised Selected Papers*. Springer, 2023, pp. 139–160. ISBN: 978-3-031-36840-0. DOI: 10.1007/978-3-031-36840-0_8. URL: https://doi.org/10.1007/978-3-031-36840-0_8.

[91] Morten T. Hansen, Nitin Nohria, and Thomas Tierney. “What’s your strategy for managing knowledge”. In: *Knowledge management: critical perspectives on business and management* 77.2 (2005), p. 322.

[92] Bhadrappa Haralayya and Munikrishnaiah Sundararamaiah. “Knowledge Management Systems in the Digital Age: Enhancing Collaboration and Decision-Making in Orginizations”. In: *Available at SSRN 5252702* (2025). URL: https://dx.doi.org/10.2139/ssrn.5252702.

[93] Tom E. Hardwicke, Stylianos Serghiou, Perrine Janiaud, Valentin Danchev, Sophia Crüwell, Steven N Goodman, and John P. A. Ioannidis. “Calibrating the scientific ecosystem through meta-research”. In: *Annual Review of Statistics and Its Application* 7.1 (2020), pp. 11–37. URL: https://doi.org/10.1146/annurev-statistics-031219-041104.

[94] Syed M. Hasan, Abdur R. Shahid, and Ahmed Imteaj. “Evaluating Sustainability and Social Costs of Adversarial Training in Machine Learning”. In: *IEEE Consumer Electronics Magazine* (2024), pp. 1–6. DOI: 10.1109/MCE.2024.3458350. URL: https://doi.org/10.1109/MCE.2024.3458350.

[95] Syed Mhamudul Hasan, Abdur R. Shahid, and Ahmed Imteaj. "Towards Sustainable SecureML: Quantifying Carbon Footprint of Adversarial Machine Learning". In: *IEEE International Conference on Communications Workshops, ICC 2024 Workshops, Denver, CO, USA, June 9-13, 2024*. IEEE, 2024, pp. 1359–1364. DOI: 10.1109/ICCWORKSHOPS59551.2024.10615723. URL: https://doi.org/10.1109/ICCWorkshops59551.2024.10615723.

[96] Stefan Haselböck, Rainer Weinreich, and Georg Buchgeher. "Decision Models for Microservices: Design Areas, Stakeholders, Use Cases, and Requirements". In: *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*. Ed. by Antónia Lopes and Rogério de Lemos. Vol. 10475. Lecture Notes in Computer Science. Springer, 2017, pp. 155–170. DOI: 10.1007/978-3-319-65831-5_11. URL: https://doi.org/10.1007/978-3-319-65831-5_11.

[97] Wilhelm Hasselbring. "Benchmarking as Empirical Standard in Software Engineering Research". In: *EASE 2021: Evaluation and Assessment in Software Engineering, Trondheim, Norway, June 21-24, 2021*. Ed. by Ruzanna Chitchyan, Jingyue Li, Barbara Weber, and Tao Yue. ACM, 2021, pp. 365–372. DOI: 10.1145/3463274.3463361. URL: https://doi.org/10.1145/3463274.3463361.

[98] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "DeBERTa: Decoding-Enhanced Bert with Disentangled Attention". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[99] Ben Hermann, Stefan Winter, and Janet Siegmund. "Community expectations for research artifacts and evaluation processes". In: *ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual Event, USA, November 8-13, 2020*. Ed. by Prem Devanbu, Myra B. Cohen, and Thomas Zimmermann. ACM, 2020, pp. 469–480. DOI: 10.1145/3368089.3409767. URL: https://doi.org/10.1145/3368089.3409767.

[100] Andrew Hodrien and Terrence Fernando. "A review of post-study and post-task subjective questionnaires to guide assessment of system usability". In: *Journal of Usability Studies* 16.3 (2021), pp. 203–232.

[101] Andreas Höfer and Walter F. Tichy. "Status of Empirical Research in Software Engineering". In: *Empirical Software Engineering Issues. Critical Assessment and Future Directions*. Springer Berlin Heidelberg, 2007, pp. 10–19. DOI: 10.1007/978-3-540-71301-2_3.

[102] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. "Knowledge Graphs". In: *arXiv* (2021). URL: https://arxiv.org/pdf/2003.02320.

[103] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. "Knowledge Graphs". In: *ACM Computing Surveys* 54.4 (2022), 71:1–71:37. DOI: 10.1145/3447772. URL: https://doi.org/10.1145/3447772.

[104] Zhi Hong, Aswathy Ajith, J. Gregory Pauloski, Eamon Duede, Carl Malamud, Roger Magoulas, Kyle Chard, and Ian T. Foster. "The diminishing returns of masked language models to science". In: *CoRR* abs/2205.11342 (2022). DOI: 10.48550/ARXIV.2205.11342. URL: https://doi.org/10.48550/arXiv.2205.11342.

[105] Xin Huang, He Zhang, Xin Zhou, Muhammad Ali Babar, and Song Yang. "Synthesizing qualitative research in software engineering: a critical review". In: *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*. Ed. by Michel Chaudron, Ivica Crnkovic, Marsha Chechik, and Mark Harman. ACM, 2018, pp. 1207–1218. DOI: 10.1145/3180155.3180235. URL: https://doi.org/10.1145/3180155.3180235.

[106] Ronald J. Hunt. "Percent agreement, Pearson's correlation, and kappa as measures of inter-examiner reliability". In: *Journal of Dental Research* 65.2 (1986), pp. 128–130.

[107] Nourhan Ibrahim, Samar Aboulela, Ahmed F. Ibrahim, and Rasha F. Kashef. "A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges". In: *Discover Artificial Intelligence* 4.1 (2024), p. 76. DOI: 10.1007/S44163-024-00175-8. URL: https://doi.org/10.1007/s44163-024-00175-8.

[108] Samuel Idowu, Daniel Strüber, and Thorsten Berger. "Asset Management in Machine Learning: State-of-research and State-of-practice". In: *ACM Computing Surveys* 55.7 (2023), 144:1–144:35. DOI: 10.1145/3543847. URL: https://doi.org/10.1145/3543847.

[109] Vinodh Ilangovan, Sören Auer, Markus Stocker, Lars Vogt, and Sanju Tiwari. *Open Research Knowledge Graph*. Cuvillier Verlag, July 5, 2024. 148 pp. ISBN: 978-3-68952-038-0. DOI: 10.3233/FAIA250216. URL: https://ebooks.iospress.nl/doi/10.3233/FAIA250216.

[110] Institute of Electrical and Electronics Engineers. *IEEE Standard Glossary of Software Engineering Terminology*. New York, NY, USA: IEEE, 1983. DOI: 10.1109/IEEESTD.1983.7435207.

[111] Institute of Electrical and Electronics Engineers. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE, Dec. 1990. DOI: 10.1109/IEEESTD.1990.101064.

[112] John P. A. Ioannidis, Daniele Fanelli, Debbie Drake Dunne, and Steven N. Goodman. "Meta-research: Evaluation and Improvement of Research Methods and Practices". In: *PLOS Biology* 13.10 (Oct. 2015), pp. 1–7. DOI: 10.1371/journal.pbio.1002264. URL: https://doi.org/10.1371/journal.pbio.1002264.

[113] Chadni Islam, Muhammad Ali Babar, and Surya Nepal. "Architecture-Centric Support for Integrating Security Tools in a Security Orchestration Platform". In: *Software Architecture - 14th European Conference, ECSA 2020, L'Aquila, Italy, September 14-18, 2020, Proceedings*. Ed. by Anton Jansen, Ivano Malavolta, Henry Muccini, Ipek Ozkaya, and Olaf Zimmermann. Vol. 12292. Lecture Notes in Computer Science. Springer, 2020, pp. 165–181. DOI: 10.1007/978-3-030-58923-3_11. URL: https://doi.org/10.1007/978-3-030-58923-3_11.

[114] International Organization for Standardization. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. Standard ISO/IEC 25010:2011. Geneva, Switzerland: ISO/IEC, Mar. 2011. URL: https://www.iso.org/standard/35733.html.

[115] Aman Jaffari, Jihyun Lee, and Eunmi Kim. "Variability Modeling in Software Product Line: A Systematic Literature Review". In: *Software Engineering in IoT, Big Data, Cloud and Mobile Computing*. Ed. by Haengkon Kim and Roger Lee. Cham: Springer International Publishing, 2021, pp. 1–15. ISBN: 978-3-030-64773-5. DOI: 10.1007/978-3-030-64773-5_1. URL: https://doi.org/10.1007/978-3-030-64773-5_1.

[116] Siddhant Jain, Asheesh Kumar, Trinita Roy, Kartik Shinde, Goutham Vignesh, and Rohan Tondulkar. "SciSpace Literature Review: Harnessing AI for Effortless Scientific Discovery". In: *Advances in Information Retrieval*. Ed. by Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis. Cham: Springer Nature Switzerland, 2024, pp. 256–260. ISBN: 978-3-031-56069-9. URL: https://doi.org/10.1007/978-3-031-56069-9_28.

[117] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. "Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge". In: *Proceedings of the 10th International Conference on Knowledge Capture*. K-CAP '19. Marina Del Rey, CA, USA: Association for Computing Machinery, 2019, pp. 243–246. ISBN: 9781450370080. DOI: 10.1145/3360901.3364435. URL: https://doi.org/10.1145/3360901.3364435.

[118] Mohamad Yaser Jaradeh, Markus Stocker, and Sören Auer. "Question Answering on Scholarly Knowledge Graphs". In: *Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25-27, 2020, Proceedings*. Ed. by Mark M. Hall, Tanja Merčun, Thomas Risse, and Fabien Duchateau. Vol. 12246. Lecture Notes in Computer Science. Springer, 2020, pp. 19–32. DOI: 10.1007/978-3-030-54956-5_2. URL: https://doi.org/10.1007/978-3-030-54956-5_2.

[119] Andreas Jedlitschka, Marcus Ciolkowski, and Dietmar Pfahl. "Reporting Experiments in Software Engineering". In: *Guide to Advanced Empirical Software Engineering*. Ed. by Forrest Shull, Janice Singer, and Dag I. K. Sjøberg. Springer, 2008, pp. 201–228. DOI: 10.1007/978-1-84800-044-5_8. URL: https://doi.org/10.1007/978-1-84800-044-5_8.

[120] Andreas Jedlitschka and Dietmar Pfahl. “Reporting guidelines for controlled experiments in software engineering”. In: *2005 International Symposium on Empirical Software Engineering (ISESE 2005), 17-18 November 2005, Noosa Heads, Australia*. IEEE Computer Society, 2005, pp. 95–104. DOI: 10.1109/ISESE.2005.1541818. URL: https://doi.org/10.1109/ISESE.2005.1541818.

[121] Yuna Jeong and Eunhui Kim. “SciDeBERTa: Learning DeBERTa for Science Technology Documents and Fine-Tuning Information Extraction Tasks”. In: *IEEE Access* 10 (2022), pp. 60805–60813. DOI: 10.1109/ACCESS.2022.3180830. URL: https://doi.org/10.1109/ACCESS.2022.3180830.

[122] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stockand Teven Le Scaoand Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. “Mistral 7B”. In: *arXiv preprint arXiv:2310.08825* 3 (2023). DOI: 10.48550/arXiv.2310.06825.

[123] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. “Mixtral of experts”. In: *arXiv preprint arXiv:2401.04088* (2024). DOI: 10.48550/arXiv.2401.04088.

[124] Boran Jiang, Yuqi Wang, Yi Luo, Dawei He, Peng Cheng, and Liangcai Gao. “Reasoning on Efficient Knowledge Paths: Knowledge Graph Guides Large Language Model for Domain Question Answering”. In: *IEEE International Conference on Knowledge Graph, ICKG 2023, Shanghai, China, December 1-2, 2023*. Ed. by Victor S. Sheng, Chindo Hicks, Charles Ling, Vijay Raghavan, and Xindong Wu. IEEE, 2024, pp. 142–149. DOI: 10.1109/ICKG63256.2024.00026. URL: https://doi.org/10.1109/ICKG63256.2024.00026.

[125] Longquan Jiang, Xi Yan, and Ricardo Usbeck. “A Structure and Content Prompt-based Method for Knowledge Graph Question Answering over Scholarly Data”. In: *Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023*. Ed. by Debayan Banerjee, Ricardo Usbeck, Nandana Mihindukulasooriya, Gunjan Singh, Raghava Mutharaju, and Pavan Kapanipathi. Vol. 3592. CEUR Workshop Proceedings. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper3.pdf.

[126] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. “Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs”. In: *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational Linguistics, 2024, pp. 163–184. DOI: 10.18653/V1/2024.FINDINGS-ACL.11. URL: https://doi.org/10.18653/v1/2024.findings-acl.11.

[127] Eugenie Evelynne Johnson, Hannah O'Keefe, Anthea Sutton, and Christopher Marshall. "The Systematic Review Toolbox: keeping up to date with tools to support evidence synthesis". In: *Systematic Reviews* 11.1 (2022), p. 258. URL: https://doi.org/10.1186/s13643-022-02122-z.

[128] Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak, and A. Spencer Peterson. *Feature-oriented domain analysis (FODA) feasibility study*. Tech. rep. 1990. URL: https://doi.org/10.21236/ADA235785.

[129] Angelika Kaplan. "Classification of Software Engineering Contributions to Support Validation Techniques and Research Applications". In: *The International Conference on Evaluation and Assessment in Software Engineering 2022*. EASE 2022. Gothenburg, Sweden: Association for Computing Machinery, 2022, pp. 323–328. ISBN: 9781450396134. DOI: 10.1145/3530019.3535345. URL: https://doi.org/10.1145/3530019.3535345.

[130] Angelika Kaplan, Fatma Chebbi, Dominik Fuchß, Oliver Karras, Tobias Hey, Anne Koziolek, and Ralf H. Reussner. "Structuring Scientific Knowledge in Software Engineering Using the Open Research Knowledge Graph: A Use Case and Experience Report". In: *Software Engineering 2025 Companion Proceedings, Fachtagung des GI-Fachbereichs Softwaretechnik, Karlsruhe, Germany, February 24-28, 2025*. Ed. by Kevin Feichtinger, Lisa Sonnleithner, and Hamideh Hajiabadi. Gesellschaft für Informatik e.V., 2025, p. 29. DOI: 10.18420/SE2025-WS-29. URL: https://doi.org/10.18420/se2025-ws-29.

[131] Angelika Kaplan, Jan Keim, Lukas Greiner, Anne Koziolek, and Ralf H. Reussner. "A Classification Framework for Scientific Documents to Support Knowledge Graph Population". In: *Software Engineering 2025 Companion Proceedings, Fachtagung des GI-Fachbereichs Softwaretechnik, Karlsruhe, Germany, February 24-28, 2025*. Ed. by Kevin Feichtinger, Lisa Sonnleithner, and Hamideh Hajiabadi. Gesellschaft für Informatik e.V., 2025, p. 28. DOI: 10.18420/SE2025-WS-28. URL: https://doi.org/10.18420/se2025-ws-28.

[132] Angelika Kaplan, Jan Keim, Lukas Greiner, Ralf Sieger, Raffaela Mirandola, and Ralf H. Reussner. "Responsible and Sustainable AI: Considering Energy Consumption in Automated Text Classification Evaluation Tasks". In: *9th IEEE/ACM International Workshop on Green and Sustainable Software, GREENS@ICSE 2025, Ottawa, ON, Canada, April 29, 2025*. IEEE, 2025, pp. 76–83. DOI: 10.1109/GREENS66463.2025.00016. URL: https://publikationen.bibliothek.kit.edu/1000180194.

[133] Angelika Kaplan, Jan Keim, Marco Schneider, Anne Koziolek, and Ralf H. Reussner. "Combining Knowledge Graphs and Large Language Models to Ease Knowledge Access in Software Architecture Research". In: *Second International Workshop on Semantic Technologies and Deep Learning Models for Scientific, Technical and Legal Data (SemTech4STLD) co-located with the Extended Semantic Web Conference 2024 (ESWC 2024), Hersonissos, Greece, May 26th, 2024*. Ed. by Rima Dessi, Danilo Dessì, Francesco Osborne, and Hidir Aras. Vol. 3697. CEUR Workshop Proceedings. CEUR-WS.org, 2024, pp. 76–82. URL: https://ceur-ws.org/Vol-3697/short3.pdf.

[134] Angelika Kaplan, Jan Keim, Marco Schneider, and Ralf H. Reussner. "HubLink: A Novel Question Answering Retrieval Approach over Knowledge Graphs". In: *Proceedings of the International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG 2025) co-located with the 24rd International Semantic Web Conference (ISWC 2025), Nara, Japan, November 2–6, 2025*. Ed. by Daniil Dobriy, Sanju Tiwari, Jennifer D'Souza, Nandana Mihindukulasooriya, and Francesco Osborne. CEUR Workshop Proceedings. CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4079/paper1.pdf.

[135] Angelika Kaplan, Marco Konersmann, Thomas Kühn, Raffaela Mirandola, and Ralf H. Reussner. "A Sustainable Data Extraction Documentation Process for Literature Studies". In: *Proceedings of the 2026 IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering, WSESE@ICSE 2026, Rio de Janeiro, Brazil, April 12-18, 2026*. ACM, 2026, pp. 70–76. DOI: 10.1145/3786149.3788301. URL: https://doi.org/10.1145/3786149.3788301.

[136] Angelika Kaplan, Thomas Kühn, Sebastian Hahner, Niko Benkler, Jan Keim, Dominik Fuchß, Sophia Corallo, and Robert Heinrich. "Introducing an Evaluation Method for Taxonomies". In: *The International Conference on Evaluation and Assessment in Software Engineering 2022*. EASE 2022. Gothenburg, Sweden: Association for Computing Machinery, 2022, pp. 311–316. ISBN: 9781450396134. DOI: 10.1145/3530019.3535305. URL: https://doi.org/10.1145/3530019.3535305.

[137] Angelika Kaplan, Thomas Kühn, and Ralf H. Reussner. *Unifying Classification Schemes for Software Engineering Meta-Research*. 2022. DOI: 10.48550/ARXIV.2209.10491. URL: https://arxiv.org/abs/2209.10491.

[138] Angelika Kaplan, Martina Rapp, Sofia Ananieva, Hamideh Hajiabad, Romina Kühn, Manar Mazkatli, Larissa Schmid, and Snigdha Singh. "Towards Managing and Organizing Research Activities". In: *8th ACM Celebration of Women in Computing: womENcourage 2021* (Online, Sept. 22–24, 2021). 2021.

[139] Angelika Kaplan, Maximilian Walter, and Robert Heinrich. "A Classification for Managing Software Engineering Knowledge". In: *Evaluation and Assessment in Software Engineering*. EASE 2021. Trondheim, Norway: Association for Computing Machinery, 2021, pp. 340–346. ISBN: 9781450390538. DOI: 10.1145/3463274.3463357. URL: https://doi.org/10.1145/3463274.3463357.

[140] Oliver Karras. "KG-EmpiRE: A Community-Maintainable Knowledge Graph for a Sustainable Literature Review on the State and Evolution of Empirical Research in Requirements Engineering". In: *32nd IEEE International Requirements Engineering Conference, RE 2024, Reykjavik, Iceland, June 24-28, 2024*. Ed. by Grischa Liebel, Irit Hadar, and Paola Spoletini. IEEE, 2024, pp. 500–501. DOI: 10.1109/RE59067.2024.00063. URL: https://doi.org/10.1109/RE59067.2024.00063.

[141] Oliver Karras, Laura Budde, Paulina Merkel, Jörg Hermsdorf, Malte Stonis, Ludger Overmeyer, Bernd-Arno Behrens, and Sören Auer. "Organizing Scientific Knowledge from Engineering Sciences Using the Open Research Knowledge Graph: The Tailored Forming Process Chain Use Case". In: *Data Science Journal* 23 (2024). DOI: 10.5334/DSJ-2024-052. URL: https://doi.org/10.5334/dsj-2024-052.

[142] Oliver Karras, Felix Wernlein, Jil Klünder, and Sören Auer. "Divide and Conquer the EmpiRE: A Community-Maintainable Knowledge Graph of Empirical Research in Requirements Engineering". In: *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2023, New Orleans, LA, USA, October 26-27, 2023*. IEEE, 2023, pp. 1–12. DOI: 10.1109/ESEM56168.2023.10304795. URL: https://doi.org/10.1109/ESEM56168.2023.10304795.

[143] Rick Kazman, Roberto Tonelli, and Cesare Pautasso. "An Empirical Basis for Software Architecture Research". In: *Software Architecture: Research Roadmaps from the Community*. Ed. by Patrizio Pelliccione, Rick Kazman, Ingo Weber, and Anna Liu. Cham: Springer Nature Switzerland, 2023, pp. 87–100. ISBN: 978-3-031-36847-9. DOI: 10.1007/978-3-031-36847-9_5. URL: https://doi.org/10.1007/978-3-031-36847-9_5.

[144] Jan Keim, Tobias Hey, Bjarne Sauer, and Anne Koziolek. "A Taxonomy for Design Decisions". In: *Software Architecture. ECSA 2022 Tracks and Workshops: Prague, Czech Republic, September 19–23, 2022, Revised Selected Papers*. Vol. 13928. Springer, 2023, pp. 439–454.

[145] Jan Keim, Sophie Schulz, Dominik Fuchß, Claudius Kocher, Janek Speit, and Anne Koziolek. "Trace Link Recovery for Software Architecture Documentation". en. In: *Software Architecture - 15th European Conference, ECSA 2021*. Ed. by Stefan Biffl, Elena Navarro, Welf Löwe, Marjan Sirjani, Raffaela Mirandola, and Danny Weyns. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 101–116. ISBN: 978-3-030-86044-8. DOI: 10.1007/978-3-030-86044-8_7.

[146] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. "KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, 2023, pp. 9410–9421. DOI: 10.18653/V1/2023.FINDINGS-EMNLP.631. URL: https://doi.org/10.18653/v1/2023.findings-emnlp.631.

[147] Barabara Kitchenham and Stuart Charters. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. [Last accessed on 2025-08-18]. 2007. URL: https://ebse.webspace.durham.ac.uk/wp-content/uploads/sites/49/2022/08/Systematic-reviews-5-8.pdf.

[148] Barbara A. Kitchenham and Pearl Brereton. "A systematic review of systematic review process research in software engineering". In: *Information and Software Technology* 55.12 (2013), pp. 2049–2075. DOI: 10.1016/J.INFSOF.2013.07.010. URL: https://doi.org/10.1016/j.infsof.2013.07.010.

[149] Barbara A. Kitchenham, Pearl Brereton, Stephen Owen, John B. Butcher, and Clive Jefferies. "Length and readability of structured software engineering abstracts". In: *IET Software* 2.1 (2008), pp. 37–45. DOI: 10.1049/IET-SEN:20070044. URL: https://doi.org/10.1049/iet-sen:20070044.

[150] Barbara A. Kitchenham, Tore Dybå, and Magne Jørgensen. “Evidence-Based Software Engineering”. In: *26th International Conference on Software Engineering (ICSE 2004), 23-28 May 2004, Edinburgh, United Kingdom*. Ed. by Anthony Finkelstein, Jacky Estublier, and David S. Rosenblum. IEEE Computer Society, 2004, pp. 273–281. DOI: 10.1109/ICSE.2004.1317449. URL: https://doi.org/10.1109/ICSE.2004.1317449.

[151] Kilian Kluge and Patrycja Jenkner. *13 Best Tools for ML Experiment Tracking and Management in 2024*. [Last accessed on 2024-11-11]. Sept. 2024. URL: https://neptune.ai/blog/best-ml-experiment-tracking-tools.

[152] Nikolai D. Kondratieff. “The long waves in economic life”. In: *Review (Fernand Braudel Center)* (1979), pp. 519–562.

[153] Marco Konersmann, Angelika Kaplan, Thomas Kühn, Robert Heinrich, Anne Koziolek, Ralf H. Reussner, Jan Jürjens, Mahmood al-Doori, Nicolas Boltz, Marco Ehl, Dominik Fuchß, Katharina Groser, Sebastian Hahner, Jan Keim, Matthias Lohr, Timur Saglam, Sophie Schulz, and Jan-Philipp Töberg. “Evaluation Methods and Replicability of Software Architecture Research Objects”. In: *19th IEEE International Conference on Software Architecture, ICSA 2022, Honolulu, HI, USA, March 12-15, 2022*. IEEE, 2022, pp. 157–168. DOI: 10.1109/ICSA53651.2022.00023. URL: https://doi.org/10.1109/ICSA53651.2022.00023.

[154] Marco Konersmann, Angelika Kaplan, Thomas Kühn, Robert Heinrich, Anne Koziolek, Ralf H. Reussner, Jan Jürjens, Mahmood al-Doori, Nicolas Boltz, Marco Ehl, Dominik Fuchß, Katharina Großer, Sebastian Hahner, Jan Keim, Matthias Lohr, Timur Saglam, Sophie Schulz, and Jan-Philipp Töberg. “Evaluation Methods and Replicability of Software Architecture Research Ojects”. In: *Software Engineering 2023, Fachtagung des GI-Fachbereichs Softwaretechnik, 20.-24. Februar 2023, Paderborn*. Ed. by Gregor Engels, Regina Hebig, and Matthias Tichy. Vol. P-332. LNI. Gesellschaft für Informatik e.V., 2023, pp. 75–76. ISBN: 978-3-88579-726-5.

[155] Marco Konersmann, Angelika Kaplan, Thomas Kühn, Robert Heinrich, Anne Koziolek, Ralf H. Reussner, Jan Jürjens, Mahmood Al-Doori, Nicolas Boltz, Marco Ehl, Dominik Fuchß, Katharina Groser, Sebastian Hahner, Jan Keim, Matthias Lohr, Timur Saglam, Sophie Schulz, and Jan-Philipp Töberg. “Replication Package of "Evaluation Methods and Replicability of Software Architecture Research Objects"”. In: *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. 2022, pp. 58–58. DOI: 10.1109/ICSA-C54293.2022.00021.

[156] Marco Konersmann, Angelika Kaplan, Thomas Kühn, Robert Heinrich, Anne Koziolek, Ralf H. Reussner, Jan Jürjens, Mahmood al-Doori, Nicolas Bolz, Marco Ehl, Dominik Fuchß, Katharina Großer, Sebastian Hahner, Jan Keim, Matthias Lohr, Timur Sağlam, Sophie Schulz, and Jan-Philipp Töberg. *Dataset for ’Evaluation Methods and Replicability of Software Architecture Research Objects’*. Zenodo, Jan. 2022. DOI: 10.5281/zenodo.6044059. URL: https://doi.org/10.5281/zenodo.6044059.

[157] Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. "HDLTex: Hierarchical Deep Learning for Text Classification". In: *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*. Ed. by Xue-wen Chen, Bo Luo, Feng Luo, Vasile Palade, and M. Arif Wani. IEEE, 2017, pp. 364–371. DOI: 10.1109/ICMLA.2017.0-134. URL: https://doi.org/10.1109/ICMLA.2017.0-134.

[158] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. "Text Classification Algorithms: A Survey". In: *Information* 10.4 (2019), p. 150. DOI: 10.3390/INFO10040150. URL: https://doi.org/10.3390/info10040150.

[159] Jan Kramer, Jan Martijn E. M. van der Werf, Johan Stokking, and Marcela Ruiz. "A Blockchain-Based Micro Economy Platform for Distributed Infrastructure Initiatives". In: *IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018*. IEEE Computer Society, 2018, pp. 11–20. DOI: 10.1109/ICSA.2018.00010. URL: https://doi.org/10.1109/ICSA.2018.00010.

[160] Klaus Krippendorff. *Content Analysis: An Introduction to its Methodology*. Sage Publishing, 2018. ISBN: 9781506395661.

[161] Klaus Krippendorff. "Reliability in content analysis: Some common misconceptions and recommendations". In: *Human Comm. Research* 30.3 (2004), pp. 411–433.

[162] Stefan Kugele, Philipp Obergfell, Manfred Broy, Oliver Creighton, Matthias Traub, and Wolfgang Hopfensitz. "On Service-Orientation for Automotive Software". In: *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 193–202. DOI: 10.1109/ICSA.2017.20. URL: https://doi.org/10.1109/ICSA.2017.20.

[163] Thomas Kühn, Dominik Fuchß, Sophie Corallo, Lars König, Erik Burger, Jan Keim, Manar Mazkatli, T. Saglam, Frederik Reiche, and Anne Koziolek. "A formalized classification schema for model consistency". In: *Karlsruher Institut für Technologie (KIT), Tech. Rep* 1 (2023). URL: https://doi.org/10.5445/IR/1000161127.

[164] Marco Kuhrmann, Daniel Méndez Fernández, and Maya Daneva. "On the pragmatic design of literature studies in software engineering: an experience-based guideline". In: *Empirical Software Engineering* 22.6 (2017), pp. 2852–2891. DOI: 10.1007/S10664-016-9492-Y. URL: https://doi.org/10.1007/s10664-016-9492-y.

[165] Dennis Kundisch, Jan Muntermann, Anna Maria Oberländer, Daniel Rau, Maximilian Röglinger, Thorsten Schoormann, and Daniel Szopinski. "An update for taxonomy designers: methodological guidance from information systems research". In: *Business & Information Systems Engineering* (2021), pp. 1–19. DOI: 10.1007/s12599-021-00723-x. URL: https://doi.org/10.1007/s12599-021-00723-x.

[166] Suchetha N. Kunnath, David Pride, and Petr Knoth. “Prompting Strategies for Citation Classification”. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*. Ed. by Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos. ACM, 2023, pp. 1127–1137. DOI: 10.1145/3583780.3615018. URL: https://doi.org/10.1145/3583780.3615018.

[167] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. “Quantifying the Carbon Emissions of Machine Learning”. In: *CoRR* abs/1910.09700 (2019). arXiv: 1910.09700. URL: http://arxiv.org/abs/1910.09700.

[168] Patricia Lago, Per Runeson, Qunying Song, and Roberto Verdecchia. “Threats to Validity in Software Engineering - hypocritical paper section or essential analysis?” In: *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2024, Barcelona, Spain, October 24-25, 2024*. Ed. by Xavier Franch, Maya Daneva, Silverio Martínez-Fernández, and Luigi Quaranta. ACM, 2024, pp. 314–324. DOI: 10.1145/3674805.3686691. URL: https://doi.org/10.1145/3674805.3686691.

[169] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=H1eA7AEtvS.

[170] J. Richard Landis and Gary G. Koch. “The measurement of observer agreement for categorical data”. In: *Biometrics* 1 (1977), pp. 159–174. DOI: doi:10.2307/2529310.

[171] Loïc Lannelongue, Jason Grealey, and Michael Inouye. “Green algorithms: quantifying the carbon footprint of computation”. In: *Advanced science* 8.12 (2021), p. 2100707.

[172] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. “Deep learning”. In: *Nat.* 521.7553 (2015), pp. 436–444. DOI: 10.1038/NATURE14539. URL: https://doi.org/10.1038/nature14539.

[173] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *CoRR* abs/1901.08746 (2019). arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746.

[174] Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. *Open Source Strikes Bread - New Fluffy Embedding Model*. [Last Accessed on 2025-09-01]. 2024. URL: https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

[175] Jens Lehmann, Antonello Meloni, Enrico Motta, Francesco Osborne, Diego Reforgiato Recupero, Angelo Antonio Salatino, and Sahar Vahdati. “Large Language Models for Scientific Question Answering: An Extensive Analysis of the SciQA Benchmark”. In: *The Semantic Web - 21st International Conference, ESWC 2024,*

Hersonissos, Crete, Greece, May 26-30, 2024, Proceedings, Part I. Ed. by Albert Meroño-Peñuela, Anastasia Dimou, Raphaël Troncy, Olaf Hartig, Maribel Acosta, Mehwish Alam, Heiko Paulheim, and Pasquale Lisena. Vol. 14664. Lecture Notes in Computer Science. Springer, 2024, pp. 199–217. DOI: 10.1007/978-3-031-60626-7_11. URL: https://doi.org/10.1007/978-3-031-60626-7_11.

[176] Thomas Leich, Sven Apel, Laura Marnitz, and Gunter Saake. "Tool support for feature-oriented software development: featureIDE: an Eclipse-based approach". In: *Proceedings of the 2005 OOPSLA workshop on Eclipse Technology eXchange, ETX 2005, San Diego, California, USA, October 16-17, 2005*. Ed. by Margaret-Anne D. Storey, Michael G. Burke, Li-Te Cheng, and André van der Hoek. ACM, 2005, pp. 55–59. DOI: 10.1145/1117696.1117708. URL: https://doi.org/10.1145/1117696.1117708.

[177] James R. Lewis. "Computer system usability questionnaire". In: *International Journal of Human-Computer Interaction* (1995).

[178] James R. Lewis. "The system usability scale: past, present, and future". In: *International Journal of Human-Computer Interaction* 34.7 (2018), pp. 577–590.

[179] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, 2020, pp. 7871–7880. URL: https://doi.org/10.18653/v1/2020.acl-main.703.

[180] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[181] Mingze Li, Haoran Yang, Zhaotai Liu, Mirza Mohtashim Alam, Harald Sack, Genet Asefa Gesese, et al. "KGMistral: Towards boosting the performance of large language models for question answering with knowledge graph integration". In: *Workshop on Deep Learning and Large Language Models for Knowledge Graphs*. 2024. URL: https://www.fiz-karlsruhe.de/sites/default/files/FIZ/Dokumente/Forschung/ISE/Publications/Conferences-Workshops/7-KGMistral-Towards-Boosting-t.pdf.

[182] Zheng Li and Austen Rainer. "Academic search engines: constraints, bugs, and recommendations". In: *Proceedings of the 13th International Workshop on Automating Test Case Design, Selection and Evaluation, A-TEST 2022, Singapore, Singapore,*

November 17-18, 2022. ACM, 2022, pp. 25–32. DOI: 10.1145/3548659.3561310. URL: https://doi.org/10.1145/3548659.3561310.

[183] Rensis Likert. *A Technique for the Measurement of Attitudes*. Monograph No. 140. New York: Archives of Psychology, Columbia University, 1932, p. 55.

[184] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (WAS 2004)*. Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.pdf.

[185] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[186] Alexandra Sasha Luccioni and Alex Hernández-García. "Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning". In: *CoRR* abs/2302.08476 (2023). DOI: 10.48550/ARXIV.2302.08476. URL: https://doi.org/10.48550/arXiv.2302.08476.

[187] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. "Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation". In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL: https://openreview.net/forum?id=oFBu7qaZpS.

[188] Cleyton V. C. de Magalhães, Fabio Q. B. da Silva, Ronnie E. S. Santos, and Marcos Suassuna. "Investigations about replication of empirical studies in software engineering: A systematic mapping study". In: *Information and Software Technology* 64 (2015), pp. 76–101. DOI: 10.1016/j.infsof.2015.02.001. URL: https://doi.org/10.1016/j.infsof.2015.02.001.

[189] Aniruddha Maiti, Samuel Adewumi, Temesgen Alemayehu Tikure, Zichun Wang, Niladri Sengupta, Anastasiia Sukhanova, and Ananya Jana. "Comparative Analysis of OpenAI GPT-4o and DeepSeek R1 for Scientific Text Categorization Using Prompt Engineering". In: *CoRR* abs/2503.02032 (2025). DOI: 10.48550/ARXIV.2503.02032. URL: https://doi.org/10.48550/arXiv.2503.02032.

[190] Christopher Marshall, Pearl Brereton, and Barbara A. Kitchenham. "Tools to support systematic reviews in software engineering: a feature analysis". In: *18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, London, England, United Kingdom, May 13-14, 2014*. Ed. by Martin J. Shepperd, Tracy Hall, and Ingunn Myrtveit. ACM, 2014, 13:1–13:10. DOI: 10.1145/2601248.2601270. URL: https://doi.org/10.1145/2601248.2601270.

[191] Manar Mazkatli, David Monschein, Johannes Grohmann, and Anne Koziolek. "Incremental Calibration of Architectural Performance Models with Parametric Dependencies". In: *2020 IEEE International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, March 16-20, 2020*. IEEE, 2020, pp. 23–34. DOI: 10.1109/ICSA47634.2020.00011. URL: https://doi.org/10.1109/ICSA47634.2020.00011.

[192] Jorge Melegati and Xiaofeng Wang. "Case Survey Studies in Software Engineering Research". In: *ESEM '20: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Bari, Italy, October 5-7, 2020*. Ed. by Maria Teresa Baldassarre, Filippo Lanubile, Marcos Kalinowski, and Federica Sarro. ACM, 2020, 6:1–6:12. DOI: 10.1145/3382494.3410683. URL: https://doi.org/10.1145/3382494.3410683.

[193] Daniel Méndez, Daniel Graziotin, Stefan Wagner, and Heidi Seibold. "Open Science in Software Engineering". In: *Contemporary Empirical Methods in Software Engineering*. Ed. by Michael Felderer and Guilherme Horta Travassos. Springer, 2020, pp. 477–501. DOI: 10.1007/978-3-030-32489-6_17. URL: https://doi.org/10.1007/978-3-030-32489-6_17.

[194] Samaneh Hosseini Moghaddam, Kelly Lyons, Cheryl Regehr, Vivek Goel, and Kaitlyn Regehr. *Towards a comprehensive taxonomy of online abusive language informed by machine leaning*. Apr. 2025. DOI: 10.48550/arXiv.2504.17653. URL: http://arxiv.org/abs/2504.17653.

[195] Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. "The Structure and Performance of an Open-Domain Question Answering System". In: *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000*. ACL, 2000, pp. 563–570. DOI: 10.3115/1075218.1075289. URL: https://aclanthology.org/P00-1071/.

[196] Michela Montesi and Patricia Lago. "Software engineering article types: An analysis of the literature". In: *Journal of Systems and Software* 81.10 (Oct. 2008), pp. 1694–1714. DOI: 10.1016/j.jss.2007.11.723.

[197] Douglas C. Montgomery. *Design and Analysis of Experiments*. 9th ed. John Wiley & Sons, 2017.

[198] Angelika Musil, Juergen Musil, Danny Weyns, and Stefan Biffl. "Continuous Adaptation Management in Collective Intelligence Systems". In: *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*. Ed. by Tomás Bures, Laurence Duchien, and Paola Inverardi. Vol. 11681. Lecture Notes in Computer Science. Springer, 2019, pp. 109–125. DOI: 10.1007/978-3-030-29983-5_8. URL: https://doi.org/10.1007/978-3-030-29983-5_8.

[199] Mojtaba Nayyeri, Zihao Wang, Mst. Mahfuja Akter, Mirza Mohtashim Alam, Md. Rashad Al Hasan Rony, Jens Lehmann, and Steffen Staab. "Integrating Knowledge Graph embedding and pretrained Language Models in Hypercomplex Spaces". In: *CoRR* abs/2208.02743 (2022). DOI: 10.48550/ARXIV.2208.02743. arXiv: 2208.02743. URL: https://doi.org/10.48550/arXiv.2208.02743.

[200] Leo A. Nefiodow and Simone Nefiodow. *The sixth Kondratieff: The new long wave in the global economy*. Leo Nefiodow, 2014.

[201] Dat Quoc Nguyen, Dai Quoc Nguyen, and Son Bao Pham. "Ripple Down Rules for question answering". In: *Semantic Web* 8.4 (2017), pp. 511–532. DOI: 10.3233/SW-150204.

[202] Robert C. Nickerson, Upkar Varshney, and Jan Muntermann. "A method for taxonomy development and its application in information systems". In: *European Journal of Information Systems* 22.3 (2013), pp. 336–359.

[203] Nils J. Nilsson. *Introduction to Machine Learning*. 1998. URL: https://ai.stanford.edu/~nilsson/MLBOOK.pdf.

[204] Luís Nunes, Nuno Santos, and António Rito Silva. "From a Monolith to a Microservices Architecture: An Approach Based on Transactional Contexts". In: *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*. Ed. by Tomás Bures, Laurence Duchien, and Paola Inverardi. Vol. 11681. Lecture Notes in Computer Science. Springer, 2019, pp. 37–52. DOI: 10.1007/978-3-030-29983-5_3. URL: https://doi.org/10.1007/978-3-030-29983-5_3.

[205] OpenAI. *03 mini model*. platform.openai.com/docs/models/o3-mini. [Last Accessed on 2025-09-01]. 2025.

[206] OpenAI. *GPT-3.5 Turbo Model*. https://platform.openai.com/docs/models/gpt-3.5-turbo. [Last accessed on 2025-10-16]. 2024.

[207] OpenAI. *GPT-4 Turbo Model Preview*. https://openai.com/index/new-embedding-models-and-api-updates/. [Last accessed on 2025-10-16]. 2024.

[208] OpenAI. *gpt-4o model*. platform.openai.com/docs/models/gpt-4o. [Last Accessed on 2025-09-01]. 2024.

[209] OpenAI. *gpt-4o-mini model*. https://platform.openai.com/docs/models/gpt-4o-mini. [Last Accessed on 2025-09-01]. 2024.

[210] OpenAI. *text-embedding-3-large*. https://platform.openai.com/docs/models/text-embedding-3-large. [Last Accessed on 2025-09-01]. 2024.

[211] Karen J. Ostergaard and Joshua D. Summers. "Development of a systematic classification and taxonomy of collaborative design activities". In: *Journal of Engineering Design* 20.1 (2009), pp. 57–81. DOI: 10.1080/09544820701499654. URL: https://doi.org/10.1080/09544820701499654.

[212] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web*. Tech. rep. Stanford infolab, 1999. URL: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.

[213] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap". In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), pp. 3580–3599. DOI: 10.1109/TKDE.2024.3352100. URL: https://doi.org/10.1109/TKDE.2024.3352100.

[214] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318. URL: https://aclanthology.org/P02-1040.pdf.

[215] Colin Paterson and Radu Calinescu. "Accurate Analysis of Quality Properties of Software with Observation-Based Markov Chain Refinement". In: *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 121–130. DOI: 10.1109/ICSA.2017.14. URL: https://doi.org/10.1109/ICSA.2017.14.

[216] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. "Graph Retrieval-Augmented Generation: A Survey". In: *arXiv preprint arXiv:2408.08921* arXiv:2408.08921 (Sept. 10, 2024). DOI: 10.48550/arXiv.2408.08921.

[217] Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo Fernandes, and João Saraiva. "Ranking programming languages by energy efficiency". In: *Science of Computer Programming* 205 (2021), p. 102609. ISSN: 0167-6423. DOI: 10.1016/j.scico.2021.102609. URL: https://www.sciencedirect.com/science/article/pii/S0167642321000022.

[218] Dewayne E. Perry. "A Unifying Theoretical Foundation for Software Engineering". In: *ISCA 20th International Conference on Software Engineering and Data Engineering (SEDE-2011) June 20-22, 2011, Imperial Palace Hotel, Las Vegas, Nevada USA*. Ed. by Rym Zalila-Wenkstern and Weili Wu. ISCA, 2011, pp. 1–6.

[219] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. "Systematic Mapping Studies in Software Engineering". In: *Electronic Workshops in Computing*. BCS Learning & Development, June 2008. DOI: 10.14236/ewic/ease2008.8.

[220] Kai Petersen and Çigdem Gencel. "Worldviews, Research Methods, and their Relationship to Validity in Empirical Software Engineering Research". In: *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, October 23-26, 2013*. IEEE Computer Society, 2013, pp. 81–89. DOI: 10.1109/IWSM-Mensura.2013.22. URL: https://doi.org/10.1109/IWSM-Mensura.2013.22.

[221] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. "Guidelines for conducting systematic mapping studies in software engineering: An update". In: *Information and Software Technology* 64 (Aug. 2015), pp. 1–18. DOI: 10.1016/j.infsof.2015.03.007.

[222] David Peterson and Aaron Panofsky. "Metascience as a Scientific Social Movement". In: *Minerva* 61.2 (June 2023), pp. 147–174. ISSN: 1573-1871. DOI: 10.1007/s11024-023-09490-3. URL: https://doi.org/10.1007/s11024-023-09490-3.

[223] Ilaria Pigazzini, Francesca Arcelli Fontana, and Andrea Maggioni. “Tool Support for the Migration to Microservice Architecture: An Industrial Case Study”. In: *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*. Ed. by Tomás Bures, Laurence Duchien, and Paola Inverardi. Vol. 11681. Lecture Notes in Computer Science. Springer, 2019, pp. 247–263. DOI: 10.1007/978-3-030-29983-5_17. URL: https://doi.org/10.1007/978-3-030-29983-5_17.

[224] Ken Power and Rebecca Wirfs-Brock. “Understanding Architecture Decisions in Context - An Industry Case Study of Architects’ Decision-Making Context”. In: *Software Architecture - 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24-28, 2018, Proceedings*. Ed. by Carlos E. Cuesta, David Garlan, and Jennifer Pérez. Vol. 11048. Lecture Notes in Computer Science. Springer, 2018, pp. 284–299. DOI: 10.1007/978-3-030-00761-4_19. URL: https://doi.org/10.1007/978-3-030-00761-4_19.

[225] Lutz Prechelt, Lloyd Montgomery, Julian Frattini, and Franz Zieris. “How (Not) To Write a Software Engineering Abstract”. In: *CoRR* abs/2506.21634 (2025). DOI: 10.48550/ARXIV.2506.21634. arXiv: 2506.21634. URL: https://doi.org/10.48550/arXiv.2506.21634.

[226] Jason Priem, Heather A. Piwowar, and Richard Orr. “OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts”. In: *CoRR* abs/2205.01833 (2022). DOI: 10.48550/ARXIV.2205.01833. URL: https://doi.org/10.48550/arXiv.2205.01833.

[227] *Prompting Techniques | Prompt Engineering Guide*. [Last accessed on 2025-04-04]. Apr. 2024. URL: https://www.promptingguide.ai/techniques.

[228] Nadia Qureshi, Muhammad Usman, and Naveed Ikram. “Evidence in software architecture, a systematic literature review”. In: *17th International Conference on Evaluation and Assessment in Software Engineering, EASE ’13, Porto de Galinhas, Brazil, April 14-16, 2013*. Ed. by Fabio Q. B. da Silva, Natalia Juristo Juzgado, and Guilherme Horta Travassos. ACM, 2013, pp. 97–106. DOI: 10.1145/2460999.2461014. URL: https://doi.org/10.1145/2460999.2461014.

[229] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21 (2020), 140:1–140:67. URL: https://jmlr.org/papers/v21/20-074.html.

[230] Sohaib Altaf Raja and Madihah Sheikh Abdul Aziz. “Development of Taxonomy for Addressing Decision-Related Challenges of Bug Prioritization and its Use for Scrum Retrospective.” In: *Pakistan Journal of Life & Social Sciences* 22.2 (2024). URL: https://www.pjlss.edu.pk/pdf_files/2024_2/14781-14812.pdf.

[231] Paul Ralph. "Toward Methodological Guidelines for Process Theories and Taxonomies in Software Engineering". In: *IEEE Transactions on Software Engineering* 45.7 (2019), pp. 712–735. DOI: 10.1109/TSE.2018.2796554. URL: https://doi.org/10.1109/TSE.2018.2796554.

[232] Paul Ralph, Sebastian Baltes, Domenico Bianculli, Yvonne Dittrich, Michael Felderer, Robert Feldt, Antonio Filieri, Carlo Alberto Furia, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara A. Kitchenham, Romain Robbes, Daniel Méndez, Jefferson Seide Molléri, Diomidis Spinellis, Miroslaw Staron, Klaas-Jan Stol, Damian A. Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, and Sira Vegas. "ACM SIGSOFT Empirical Standards". In: *CoRR* abs/2010.03525 (2020). arXiv: 2010.03525. URL: https://arxiv.org/abs/2010.03525.

[233] Paul Ralph and Romain Robbes. "The ACM SIGSOFT Paper and Peer Review Quality Initiative: Status Report". In: *ACM SIGSOFT Software Engineering Notes* 45.2 (2020), pp. 17–18. DOI: 10.1145/3385678.3385681. URL: https://doi.org/10.1145/3385678.3385681.

[234] Shiyali Ramamrita Ranganathan. *Prolegomena to library classification*. Madras Library Association, 1937.

[235] Simmi K. Ratan, Tanu Anand, and John Ratan. "Formulation of research question - stepwise approach". en. In: *Journal of Indian Association of Pediatric Surgeons* 24.1 (Jan. 2019), pp. 15–20. DOI: 10.4103/jiaps.JIAPS_76_18.

[236] Frederik Reiche, Ralf H. Reussner, and Robert Heinrich. "Detecting Information Flow Security Vulnerabilities by Analysis Coupling". In: *IEEE Transactions on Software Engineering* (2025). URL: https://ieeexplore.ieee.org/abstract/document/11082015/.

[237] Lozana Rossenova, Fidan Limani, Carsten Fortmann-Grote, Angelika Kaplan, Florian Thiery, Atif Latif, Renat Shigapov, Benjamin Zapilko, Moritz Schubotz, Heike Fliegl, and Tabea Tietz. "How are NFDI consortia using Knowledge Graphs? An overview of common functions and challenges". In: *The 2nd Conference on Research Data Infrastructure (CoRDI)*. Accessed: 2025-09-27. Section (Meta)data, Terminologies, Provenience / Working Group "Knowledge Graphs". 2025. URL: https://zenodo.org/records/16736078.

[238] Zhyar Rzgar K. Rostam and Gábor Kertész. *Fine-Tuning Large Language Models for Scientific Text Classification: A Comparative Study*. 2025. arXiv: 2412.00098 [cs.CL]. URL: https://arxiv.org/abs/2412.00098.

[239] Kiana Rostami, Robert Heinrich, Axel Busch, and Ralf H. Reussner. "Architecture-Based Change Impact Analysis in Information Systems and Business Processes". In: *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 179–188. DOI: 10.1109/ICSA.2017.17. URL: https://doi.org/10.1109/ICSA.2017.17.

[240] Jennifer Rowley. "The wisdom hierarchy: representations of the DIKW hierarchy". In: *Journal of Information Science* 33.2 (2007), pp. 163–180. URL: https://doi.org/10.1177/0165551506070706.

[241] Trinita Roy, Asheesh Kumar, Daksh Raghuvanshi, Siddhant Jain, Goutham Vignesh, Kartik Shinde, and Rohan Tondulkar. "SciSpace Copilot: Empowering Researchers through Intelligent Reading Assistance". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.21 (Mar. 2024), pp. 23826–23828. DOI: 10.1609/aaai.v38i21.30578. URL: https://ojs.aaai.org/index.php/AAAI/article/view/30578.

[242] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. *Case Study Research in Software Engineering: Guidelines and Examples*. 1st. Wiley Publishing, 2012. ISBN: 1118104358.

[243] Per Runeson and Martin Höst. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering". In: *Empirical Software Engineering* 14.2 (2009), pp. 131–164. DOI: 10.1007/s10664-008-9102-8. URL: https://doi.org/10.1007/s10664-008-9102-8.

[244] Nayan B. Ruparelia. "Software development lifecycle models". In: *SIGSOFT Software Engineering Notes* 35.3 (May 2010), pp. 8–13. ISSN: 0163-5948. DOI: 10.1145/1764810.1764814. URL: https://doi.org/10.1145/1764810.1764814.

[245] David L Sackett. "Evidence-based medicine". In: *Seminars in perinatology*. Vol. 21. 1. Elsevier. 1997, pp. 3–5.

[246] Mobashir Sadat and Cornelia Caragea. "Hierarchical Multi-Label Classification of Scientific Documents". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, 2022, pp. 8923–8937. DOI: 10.18653/V1/2022.EMNLP-MAIN.610. URL: https://doi.org/10.18653/v1/2022.emnlp-main.610.

[247] Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. "ScienceQA: a novel resource for question answering on scholarly articles". In: *International Journal on Digital Libraries* 23.3 (2022), pp. 289–301. DOI: 10.1007/S00799-022-00329-Y. URL: https://doi.org/10.1007/s00799-022-00329-y.

[248] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[249] Adrian Santos, Omar S. Gómez, and Natalia Juristo. "Analyzing Families of Experiments in SE: A Systematic Mapping Study". In: *IEEE Trans. Software Eng.* 46.5 (2020), pp. 566–583. DOI: 10.1109/TSE.2018.2864633. URL: https://doi.org/10.1109/TSE.2018.2864633.

[250] Adrian Santos and Natalia Juristo. "Comparing techniques for aggregating interrelated replications in software engineering". In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2018, Oulu, Finland, October 11-12, 2018*. Ed. by Markku Oivo, Daniel Méndez Fernández, and Audris Mockus. ACM, 2018, 8:1–8:10. DOI: 10.1145/3239235.3239239. URL: https://doi.org/10.1145/3239235.3239239.

[251] Joanna C. S. Santos, Anthony Peruma, Mehdi Mirakhorli, Matthias Galster, Jairo Veloz Vidal, and Adriana Sejfia. "Understanding Software Vulnerabilities Related to Architectural Security Tactics: An Empirical Investigation of Chromium, PHP and Thunderbird". In: *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 69–78. DOI: 10.1109/ICSA.2017.39. URL: https://doi.org/10.1109/ICSA.2017.39.

[252] Vinicius dos Santos, Anderson Yoshiaki Iwazaki, Kátia Romero Felizardo, Érica Ferreira de Souza, and Elisa Yumi Nakagawa. "Towards Sustainability of Systematic Literature Reviews". In: *ESEM '21: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Bari, Italy, October 11-15, 2021*. Ed. by Filippo Lanubile, Marcos Kalinowski, and Maria Teresa Baldassarre. ACM, 2021, 34:1–34:6. DOI: 10.1145/3475716.3484192. URL: https://doi.org/10.1145/3475716.3484192.

[253] Cezar Sas and Andrea Capiluppi. *Automatic Bottom-Up Taxonomy Construction: A Software Application Domain Study*. Sept. 2024. DOI: 10.48550/arXiv.2409.15881. URL: http://arxiv.org/abs/2409.15881.

[254] Guus Schreiber. "Knowledge Engineering". In: *Foundations of Artificial Intelligence* 3 (2008), pp. 929–946. URL: https://doi.org/10.1016/S1574-6526(07)03025-8.

[255] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson C. Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Miserlis Hoyle, and Philip Resnik. "The Prompt Report: A Systematic Survey of Prompting Techniques". In: *CoRR* abs/2406.06608 (2024). DOI: 10.48550/ARXIV.2406.06608. URL: https://doi.org/10.48550/arXiv.2406.06608.

[256] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. "Green AI". In: *CoRR* abs/1907.10597 (2019). arXiv: 1907.10597. URL: http://arxiv.org/abs/1907.10597.

[257] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. "Green AI". In: *Communications of the ACM* 63.12 (2020), pp. 54–63. DOI: 10.1145/3381831. URL: https://doi.org/10.1145/3381831.

[258] William A. Scott. "Reliability of content analysis: The case of nominal scale coding". In: *Public opinion quarterly* (1955), pp. 321–325.

[259] Chirag Shah, Ryen White, Reid Andersen, Georg Buscher, Scott Counts, Sarkar Das, Ali Montazer, Sathish Manivannan, Jennifer Neville, Nagu Rangan, Tara Safavi, Siddharth Suri, Mengting Wan, Leijie Wang, and Longqi Yang. "Using Large Language Models to Generate, Validate, and Apply User Intent Taxonomies". en. In: *ACM Transactions on the Web* 19.3 (Aug. 2025), pp. 1–29. ISSN: 1559-1131, 1559-114X. DOI: 10.1145/3732294. URL: https://dl.acm.org/doi/10.1145/3732294.

[260] Mary Shaw. “The Coming-of-Age of Software Architecture Research”. In: *Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, 12-19 May 2001, Toronto, Ontario, Canada*. Ed. by Hausi A. Müller, Mary Jean Harrold, and Wilhelm Schäfer. IEEE Computer Society, 2001, pp. 656–664. DOI: 10.1109/ICSE.2001.919142.

[261] Mary Shaw. “Writing Good Software Engineering Research Papers: Minitutorial”. In: *Proceedings of the 25th International Conference on Software Engineering*. ICSE ’03. Portland, Oregon: IEEE Computer Society, 2003, pp. 726–736. ISBN: 076951877X. DOI: 10.1109/ICSE.2003.1201262. URL: https://doi.org/10.1109/ICSE.2003.1201262.

[262] Haroon Sheikh, Corien Prins, and Erik Schrijvers. “Artificial Intelligence: Definition and Background”. In: *Mission AI: The New System Technology*. Cham: Springer International Publishing, 2023, pp. 15–41. ISBN: 978-3-031-21448-6. DOI: 10.1007/978-3-031-21448-6_2. URL: https://doi.org/10.1007/978-3-031-21448-6_2.

[263] Martin Shepperd, Nemitari Ajienka, and Steve Counsell. “The role and value of replication in empirical software engineering results”. In: *Information and Software Technology* 99 (July 2018), pp. 120–132. DOI: 10.1016/j.infsof.2018.01.006.

[264] Fabio Q. B. da Silva, Andre Luís de Medeiros Santos, Sérgio Soares, A. César C. França, Cleviton V. F. Monteiro, and Felipe Farias Maciel. “Six years of systematic literature reviews in software engineering: An updated tertiary study”. In: *Information and Software Technology* 53.9 (2011), pp. 899–913. DOI: 10.1016/J.INFSOF.2011.04.004. URL: https://doi.org/10.1016/j.infsof.2011.04.004.

[265] Fabio Q. B. da Silva, Marcos Suassuna, A. César C. França, Alicia M. Grubb, Tatiana B. Gouveia, Cleviton V. F. Monteiro, and Igor Ebrahim dos Santos. “Replication of empirical studies in software engineering research: a systematic mapping study”. In: *Empirical Software Engineering* 19.3 (2014), pp. 501–557. DOI: 10.1007/s10664-012-9227-7. URL: https://doi.org/10.1007/s10664-012-9227-7.

[266] Susan Elliott Sim, Steve M. Easterbrook, and Richard C. Holt. “Using Benchmarking to Advance Research: A Challenge to Software Engineering”. In: *Proceedings of the 25th International Conference on Software Engineering, May 3-10, 2003, Portland, Oregon, USA*. Ed. by Lori A. Clarke, Laurie Dillon, and Walter F. Tichy. IEEE Computer Society, 2003, pp. 74–83. DOI: 10.1109/ICSE.2003.1201189. URL: https://doi.org/10.1109/ICSE.2003.1201189.

[267] Dag I. K. Sjøberg, Tore Dybå, and Magne Jørgensen. “The Future of Empirical Methods in Software Engineering Research”. In: *International Conference on Software Engineering, ISCE 2007, Workshop on the Future of Software Engineering, FOSE 2007, May 23-25, 2007, Minneapolis, MN, USA*. Ed. by Lionel C. Briand and Alexander L. Wolf. IEEE Computer Society, 2007, pp. 358–378. DOI: 10.1109/FOSE.2007.30. URL: https://doi.org/10.1109/FOSE.2007.30.

[268] *Sklearn.Metrics.Hamming_loss — Scikit-Learn 1.4.2 Documentation*. [Last accessed on 2024-04-09]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html.

[269] *Sklearn.Metrics.Jaccard_score — Scikit-Learn 1.7.2 Documentation.* [Last accessed on 2025-10-15]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html.

[270] Darja Smite, Claes Wohlin, Zane Galvina, and Rafael Prikladnicki. "An empirically based terminology and taxonomy for global software engineering". In: *Empirical Software Engineering* 19.1 (2014), pp. 105–153. DOI: 10.1007/S10664-012-9217-9. URL: https://doi.org/10.1007/s10664-012-9217-9.

[271] Alan Snyder. "How to Get Your Paper Accepted at OOPSLA". In: *ACM SIGPLAN Notices* 26.11 (Nov. 1991), pp. 359–363. ISSN: 0362-1340. DOI: 10.1145/118014.117983. URL: https://doi.org/10.1145/118014.117983.

[272] Herbert Stachowiak. *Allgemeine Modelltheorie*. 1973.

[273] Klaas-Jan Stol and Brian Fitzgerald. "The ABC of Software Engineering Research". In: *ACM Transactions on Software Engineering and Methodology* 27.3 (2018), 11:1–11:51. DOI: 10.1145/3241743. URL: https://doi.org/10.1145/3241743.

[274] Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. "FiDeLiS: Faithful Reasoning in Large Language Models for Knowledge Graph Question Answering". In: *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, 2025, pp. 8315–8330. URL: https://aclanthology.org/2025.findings-acl.436/.

[275] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. "Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph". In: vol. abs/2307.07697. 2023. DOI: 10.48550/ARXIV.2307.07697. arXiv: 2307.07697. URL: https://doi.org/10.48550/arXiv.2307.07697.

[276] Lei Sun, Zhengwei Tao, Youdi Li, and Hiroshi Arakawa. "ODA: Observation-Driven Agent for integrating LLMs and Knowledge Graphs". In: *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational Linguistics, 2024, pp. 7417–7431. DOI: 10.18653/V1/2024.FINDINGS-ACL.442. URL: https://doi.org/10.18653/v1/2024.findings-acl.442.

[277] Daniel Szopinski, Thorsten Schoormann, and Dennis Kundisch. "Because Your Taxonomy is Worth IT: Towards a Framework for Taxonomy Evaluation". In: *27th European Conference on Information Systems (ECIS)*. Association for Information Systems, 2019, p. 19. URL: https://aisel.aisnet.org/ecis2019_rp/104.

[278] Daniel Szopinski, Thorsten Schoormann, and Dennis Kundisch. "Criteria as a Prelude for Guiding Taxonomy Evaluation". In: *53rd Hawaii International Conference on System Sciences (HICSS)*. ScholarSpace, 2020, pp. 1–10. DOI: 10.24251/HICSS.2020.622.

[279] Tilahun Abedissa Taffa and Ricardo Usbeck. "Leveraging LLMs in Scholarly Knowledge Graph Question Answering". In: *Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023*. Ed. by Debayan Banerjee, Ricardo Usbeck, Nandana Mihindukulasooriya, Gunjan Singh, Raghava Mutharaju, and Pavan Kapanipathi. Vol. 3592. CEUR Workshop Proceedings. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper5.pdf.

[280] Isabel Taverniers, Marc De Loose, and Erik Van Bockstaele. "Trends in quality in the analytical laboratory. II. Analytical method validation and quality assurance". In: *TrAC Trends in Analytical Chemistry* 23.8 (2004), pp. 535–552. ISSN: 0165-9936. DOI: https://doi.org/10.1016/j.trac.2004.04.001. URL: https://www.sciencedirect.com/science/article/pii/S0165993604030031.

[281] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. "Galactica: A Large Language Model for Science". In: *CoRR* abs/2211.09085 (2022). DOI: 10.48550/ARXIV.2211.09085. arXiv: 2211.09085. URL: https://doi.org/10.48550/arXiv.2211.09085.

[282] Qwen Team. *Qwen2.5: A Party of Foundation Models*. [Last Accessed on 2025-09-01]. Sept. 2024. URL: https://qwenlm.github.io/blog/qwen2.5/.

[283] Christopher Theisen, Marcel Dunaiski, Laurie Williams, and Willem Visser. "Software Engineering Research at the International Conference on Software Engineering in 2016". In: *ACM SIGSOFT Software Engineering Notes* 42.4 (2018), pp. 1–7. DOI: 10.1145/3149485.3149496. URL: https://doi.org/10.1145/3149485.3149496.

[284] Romal Thoppilan et al. "LaMDA: Language Models for Dialog Applications". In: *CoRR* abs/2201.08239 (2022). arXiv: 2201.08239. URL: https://arxiv.org/abs/2201.08239.

[285] Walter Tichy. "Workings of science: How software engineering research became empirical". In: *Ubiquity* 2022.July (Aug. 2022). DOI: 10.1145/3512339. URL: https://doi.org/10.1145/3512339.

[286] Jaco du Toit, Herman Redelinghuys, and Marcel Dunaiski. "Introducing Three New Benchmark Datasets for Hierarchical Text Classification". In: *CoRR* abs/2411.19119 (2024). DOI: 10.48550/ARXIV.2411.19119. arXiv: 2411.19119. URL: https://doi.org/10.48550/arXiv.2411.19119.

[287] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. "LLaMA: Open and Efficient Foundation Language Models". In: *CoRR* abs/2302.13971 (2023). DOI: 10.48550/ARXIV.2302.13971. URL: https://doi.org/10.48550/arXiv.2302.13971.

[288] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023). DOI: 10.48550/arXiv.2307.09288.

[289] Tristan Trébaol. *CUMULATOR—a tool to quantify and report the carbon footprint of machine learning computations and communication in academia and healthcare*. 2020.

[290] Cristina Tripon. "Information Literacy in Engineering Education". In: *Digital Equity and Literacy*. Ed. by Associate Prof. Sandro Serpa and Prof. Ana Isabel Santos. London: IntechOpen, 2025. Chap. 4. DOI: 10.5772/intechopen.1012562. URL: https://doi.org/10.5772/intechopen.1012562.

[291] Lewis Tunstall, Leandro von Werra, Thomas Wolf, and Aurélien Géron. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. Revised edition. Sebastopol: O'Reilly, 2022. ISBN: 978-1-09-813679-6.

[292] Muhammad Usman, Ricardo Britto, Jürgen Börstler, and Emilia Mendes. "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method". In: *Information and Software Technology* 85 (2017), pp. 43–59. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2017.01.006. URL: https://www.sciencedirect.com/science/article/pii/S0950584917300472.

[293] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[294] Roberto Verdecchia, Emelie Engström, Patricia Lago, Per Runeson, and Qunying Song. "Threats to validity in software engineering research: A critical reflection". In: *Information and Software Technology* 164 (2023), p. 107329. DOI: 10.1016/J.INFSOF.2023.107329. URL: https://doi.org/10.1016/j.infsof.2023.107329.

[295] Iris Vessey, V. Ramesh, and Robert L. Glass. "A unified classification system for research in the computing disciplines". In: *Information and Software Technology* 47.4 (2005), pp. 245–255. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2004.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0950584904001260.

[296] Lev S. Vygotsky. *Thought and Language*. MIT Press, 1962. ISBN: 9780262720014.

[297] Alex D. Wade. "The Semantic Scholar Academic Graph (S2AG)". In: *Companion Proceedings of the Web Conference 2022*. WWW '22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, p. 739. ISBN: 9781450391306. DOI: 10.1145/3487553.3527147. URL: https://doi.org/10.1145/3487553.3527147.

[298] Stefan Wagner, Marvin Muñoz Barón, Davide Falessi, and Sebastian Baltes. "Towards Evaluation Guidelines for Empirical Studies involving LLMs". In: *IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering, WSESE@ICSE 2025, Ottawa, ON, Canada, May 3, 2025*. IEEE/ACM, 2025, pp. 24–27. DOI: 10.1109/WSESE66602.2025.00011. URL: https://doi.org/10.1109/WSESE66602.2025.00011.

[299] Liangguo Wang, Ruoyu Chen, and Li Li. "Knowledge-guided prompt learning for few-shot text classification". In: *Electronics* 12.6 (2023), p. 1486.

[300] Meng Wang, Zhixiong Zhang, Hanyu Li, and Guangyin Zhang. "An Improved Meta-Knowledge Prompt Engineering Approach for Generating Research Questions in Scientific Literature". In: *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2024, Volume 1: KDIR, Porto, Portugal, November 17-19, 2024*. Ed. by Frans Coenen, Ana Fred, and Jorge Bernardino. SCITEPRESS, 2024, pp. 457–464. DOI: 10.5220/0013060900003838. URL: https://doi.org/10.5220/0013060900003838.

[301] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 176–194. DOI: 10.1162/TACL_A_00360. URL: https://doi.org/10.1162/tacl_a_00360.

[302] Yilin Wen, Zifeng Wang, and Jimeng Sun. "MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 10370–10388. DOI: 10.18653/v1/2024.acl-long.558. URL: https://aclanthology.org/2024.acl-long.558/.

[303] George R. Wheaton. *Development of a taxonomy of human performance: A review of classificatory systems relating to tasks and performance*. 1968. URL: https://apps.dtic.mil/sti/html/tr/AD0689411/.

[304] Roel J. Wieringa. "Towards a unified checklist for empirical research in software engineering: first proposal". In: *16th International Conference on Evaluation & Assessment in Software Engineering, EASE 2012, Ciudad Real, Spain, May 14-15, 2012. Proceedings*. Ed. by Maria Teresa Baldassarre, Marcela Genero, Emilia Mendes, and Mario Piattini. IET - The Institute of Engineering and Technology / IEEE Xplore, 2012, pp. 161–165. DOI: 10.1049/ic.2012.0020. URL: https://doi.org/10.1049/ic.2012.0020.

[305] Roel J. Wieringa, Neil A. M. Maiden, Nancy R. Mead, and Colette Rolland. "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion". In: *Requirements Engineering* 11.1 (2006), pp. 102–107. DOI: 10.1007/S00766-005-0021-6. URL: https://doi.org/10.1007/s00766-005-0021-6.

[306] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific data* 3.1 (2016), pp. 1–9. URL: https://doi.org/10.1038/sdata.2016.18.

[307] Robert Winter and Joachim Schelp. "Reference modeling and method construction: a design science perspective". In: *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*. Ed. by Hisham Haddad. ACM, 2006, pp. 1561–1562. DOI: 10.1145/1141277.1141638. URL: https://doi.org/10.1145/1141277.1141638.

[308] Stefan Winter, Christopher Steven Timperley, Ben Hermann, Jürgen Cito, Jonathan Bell, Michael Hilton, and Dirk Beyer. "A retrospective study of one decade of artifact evaluations". In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, Singapore, Singapore, November 14-18, 2022*. Ed. by Abhik Roychoudhury, Cristian Cadar, and Miryung Kim. ACM, 2022, pp. 145–156. DOI: 10.1145/3540250.3549172. URL: https://doi.org/10.1145/3540250.3549172.

[309] Christopher Wissuchek and Patrick Zschech. "Survey and Systematization of Prescriptive Analytics Systems: Towards Archetypes from a Human-Machine-Collaboration Perspective". In: *31st European Conference on Information Systems - Co-creating Sustainable Digital Futures, ECIS 2023, Kristiansan, Norway, June 11-16, 2023*. Ed. by Margunn Aanestad, Stefan Klein, Monideepa Tarafdar, Shengnan Han, Sven Laumer, and Isabel Ramos. Association for Information Systems (AIS), 2023. URL: https://aisel.aisnet.org/ecis2023_rp/289.

[310] Claes Wohlin. "Case Study Research in Software Engineering - It is a Case, and it is a Study, but is it a Case Study?" In: *Information and Software Technology* 133 (2021), p. 106514. DOI: 10.1016/J.INFSOF.2021.106514. URL: https://doi.org/10.1016/j.infsof.2021.106514.

[311] Claes Wohlin. *Experimentation in Software Engineering: An Introduction*. Boston: Kluwer Academic, 2000. ISBN: 0792386825.

[312] Claes Wohlin. "Writing for synthesis of evidence in empirical software engineering". In: *2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, Torino, Italy, September 18-19, 2014*. Ed. by Maurizio Morisio, Tore Dybå, and Marco Torchiano. ACM, 2014, 46:1–46:4. DOI: 10.1145/2652524.2652559. URL: https://doi.org/10.1145/2652524.2652559.

[313] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-29044-2.

[314] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. "Carbontracker: tracking and predicting the carbon footprint of training deep learning models". In: *arXiv e-prints* (2020), arXiv–2007. URL: https://doi.org/10.48550/arXiv.2007.03051.

[315] Hyrum K. Wright, Miryung Kim, and Dewayne E. Perry. “Validity concerns in software engineering research”. In: *Proceedings of the Workshop on Future of Software Engineering Research, FoSER 2010, at the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2010, Santa Fe, NM, USA, November 7-11, 2010*. Ed. by Gruia-Catalin Roman and Kevin J. Sullivan. ACM, 2010, pp. 411–414. DOI: 10.1145/1882362.1882446. URL: https://doi.org/10.1145/1882362.1882446.

[316] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. “Retrieval-Augmented Generation for Natural Language Processing: A Survey”. In: *CoRR* abs/2407.13193 (2024). DOI: 10.48550/ARXIV.2407.13193. arXiv: 2407.13193. URL: https://doi.org/10.48550/arXiv.2407.13193.

[317] Marvin Wyrich and Justus Bogner. “Beyond Self-Promotion: How Software Engineering Research Is Discussed on LinkedIn”. In: *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS2024, Lisbon, Portugal, April 14-20, 2024*. ACM, 2024, pp. 85–95. DOI: 10.1145/3639475.3640113. URL: https://doi.org/10.1145/3639475.3640113.

[318] Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. “Generate-on-Graph: Treat LLM as both Agent and KG for Incomplete Knowledge Graph Question Answering”. In: (Nov. 2024). Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, pp. 18410–18430. DOI: 10.18653/v1/2024.emnlp-main.1023. URL: https://aclanthology.org/2024.emnlp-main.1023/.

[319] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. “SGM: Sequence Generation Model for Multi-label Classification”. In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Association for Computational Linguistics, 2018, pp. 3915–3926. URL: https://aclanthology.org/C18-1330/.

[320] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 5754–5764. URL: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[321] Dilek Yargan and Ludger Jansen. “Terminological Resources for Biologically Inspired Design and Biomimetics: Evaluation of the Potential for Ontology Reuse”. In: *Biomimetics* 10.1 (2025), p. 39. DOI: 10.3390/biomimetics10010039. URL: https://www.mdpi.com/2313-7673/10/1/39.

[322] Rajitha Yasaweerasinghelage, Mark Staples, Ingo Weber, and Hye-Young Paik. "Predicting the Performance of Privacy-Preserving Data Analytics Using Architecture Modelling and Simulation". In: *IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018*. IEEE Computer Society, 2018, pp. 166–175. DOI: 10.1109/ICSA.2018.00026. URL: https://doi.org/10.1109/ICSA.2018.00026.

[323] Robert K. Yin. *Case Study Research and Applications*. Sage Publications Ltd., Jan. 2017. ISBN: 1506336167.

[324] Zhiwen You, Kanyao Han, Haotian Zhu, Bertram Ludäscher, and Jana Diesner. "SciPrompt: Knowledge-augmented Prompting for Fine-grained Categorization of Scientific Topics". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Association for Computational Linguistics, 2024, pp. 6087–6104. DOI: 10.18653/V1/2024.EMNLP-MAIN.350. URL: https://doi.org/10.18653/v1/2024.emnlp-main.350.

[325] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. "Evaluation of Retrieval-Augmented Generation: A Survey". In: *Big Data*. Singapore: Springer Nature Singapore, 2025, pp. 102–120. ISBN: 978-981-96-1024-2. DOI: 10.1007/978-981-96-1024-2_8. URL: https://doi.org/10.1007/978-981-96-1024-2_8.

[326] Muhamad Saiful Bahri Yusoff. "ABC of content validation and content validity index calculation". In: *Education in medicine journal* 11.2 (2019), pp. 49–54. URL: https://doi.org/10.21315/eimj2019.11.2.6.

[327] Carmen Zannier, Grigori Melnik, and Frank Maurer. "On the success of empirical studies in the international conference on software engineering". In: *Proceedings of the 28th International Conference on Software Engineering*. ACM, May 2006. DOI: 10.1145/1134285.1134333.

[328] Marvin V. Zelkowitz and Dolores Wallace. "Experimental Validation in Software Engineering". In: *Information and Software Technology* 39.11 (1997), pp. 735–743. ISSN: 0950-5849. DOI: 10.1016/S0950-5849(97)00025-6. URL: https://www.sciencedirect.com/science/article/pii/S0950584997000256.

[329] Haiqi Zhang, Zhengyuan Zhu, Zeyu Zhang, and Chengkai Li. "LLMTaxo: Leveraging Large Language Models for Constructing Taxonomy of Factual Claims from Social Media". In: *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, 2025, pp. 19627–19641. URL: https://aclanthology.org/2025.findings-acl.1007/.

[330] He Zhang, Muhammad Ali Babar, and Paolo Tell. "Identifying relevant studies in software engineering". In: *Information and Software Technology* 53.6 (2011). Special Section: Best papers from the APSEC, pp. 625–637. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2010.12.010. URL: https://www.sciencedirect.com/science/article/pii/S0950584910002260.

[331] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[332] Yukun Zhang, Bo Liu, Liyun Dai, Kang Chen, and Xuelian Cao. "Automated Microservice Identification in Legacy Systems with Functional and Non-Functional Metrics". In: *2020 IEEE International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, March 16-20, 2020*. IEEE, 2020, pp. 135–145. DOI: 10.1109/ICSA47634.2020.00021. URL: https://doi.org/10.1109/ICSA47634.2020.00021.

[333] Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He. "PretrainKGE: Learning Knowledge Representation from Pretrained Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Vol. EMNLP 2020. Findings of ACL. Association for Computational Linguistics, 2020, pp. 259–266. DOI: 10.18653/V1/2020.FINDINGS-EMNLP.25. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.25.

[334] Chaim Zins. "Conceptual approaches for defining data, information, and knowledge". In: *Journal of the American Society for Information Science and Technology* 58.4 (2007), pp. 479–493. URL: https://doi.org/10.1002/asi.20508.

[335] Paul G. Zurkowski. *The Information Service Environment Relationships and Priorities. Related Paper No. 5*. ERIC, 1974. URL: https://eric.ed.gov/?id=eD100391.

A. Appendix

A.1. Contribution C3: Artifact Availability of Case Studies

Table A.1.: Papers making input data of case studies available [153]

Research Object	Property	Paper ID
Architecture Analysis Method	Accuracy	[30, 215, 145]
	Effectiveness	[223, 77, 145]
	Functional Suitability	[67, 198, 145]
	Performance Efficiency	[15]
	Satisfaction	[67, 198]
Architecture Description Language	Effectiveness	[11]
	Functional Suitability	[11]
	Performance Efficiency	[11]
	Satisfaction	[11]
Architectural Aspects	Functional Suitability	[204]
	None	[77]
Architecture Optimization Method	Functional Suitability	[204]
	Satisfaction	[7]
Architectural Assumptions	Performance Efficiency	[322]
Architecture Design Method	Accuracy	[332]
Architecture Extraction	Effectiveness	[223]
Architecture Pattern	Security	[251]
Reference Architecture	Functional Suitability	[113]

A.2. Contribution C5: ORKG Template Variants

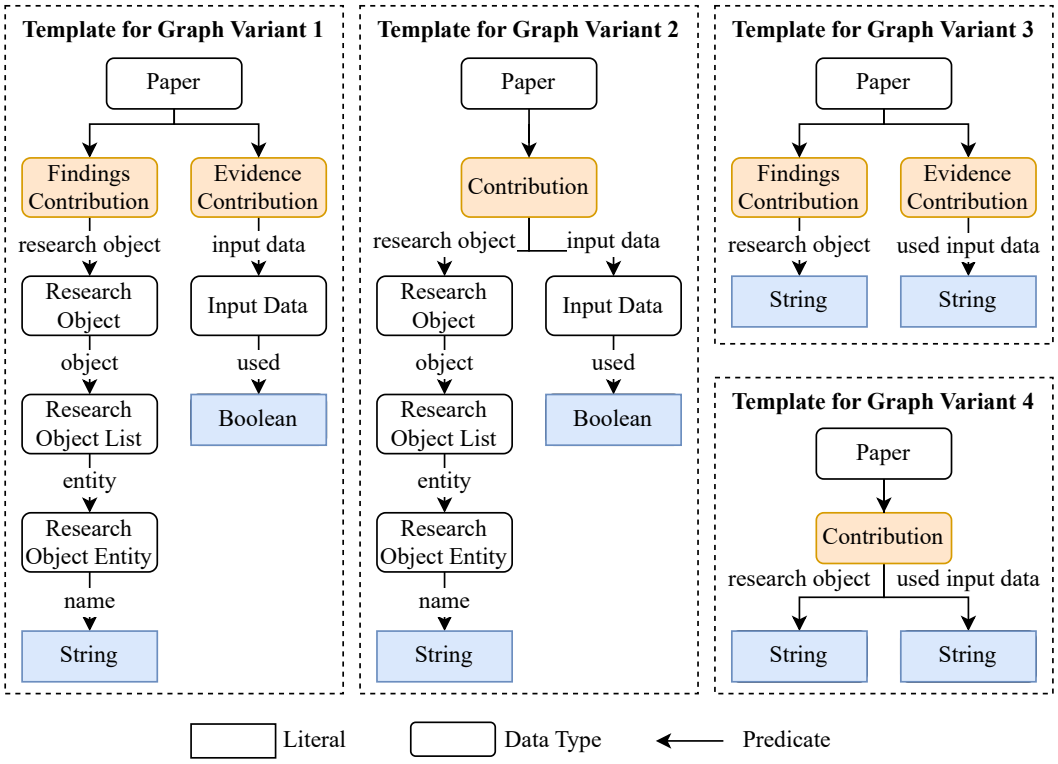

Figure A.1.: Excerpt of the four ORKG templates to derive different graph variants, featuring the *Research Object* (as part of *Findings*) and *Input Data* (as part of *Evidence*) to model knowledge-based statements. This figure highlights the difference between contribution granularity (orange) and the path length (blue).

A.3. Contribution C6: Additional Results for Classification

A.3.1. Evaluation Results for Hyperparameter Search

Table A.2.: Best configuration results for the hyperparameter search

Language Model	Category	Augm.	*ep*	*lr*	*wd*	*seed*	*bs*	F_1
BERT	Research Object	False	15	1.022e-06	5.251e-05	42172615	32	0.16
		True	10	1.422e-06	8.784e-06	804878275	16	0.19
	Evaluation Method	False	24	2.479e-06	1.597e-06	219556809	32	0.17
		True	32	3.903e-06	2.183e-05	906400257	16	0.14
	Property	False	23	7.923e-05	5.090e-06	545862785	16	0.58
		True	14	1.419e-06	1.934e-06	402962483	16	0.65
DeBERTa	Research Object	False	73	2.914e-06	7.690e-05	505504290	2	0.24
		True	25	5.596e-05	4.980e-05	448011207	8	0.27
	Evaluation Method	False	26	1.817e-06	4.268e-05	505484641	32	0.15
		True	36	1.412e-05	9.914e-06	942094705	2	0.20
	Property	False	87	3.648e-05	1.156e-06	639618901	32	0.74
		True	23	6.998e-06	1.317e-05	560303621	32	0.82
DistilBERT	Research Object	False	31	2.016e-06	1.957e-05	38883714	32	0.20
		True	92	8.671e-05	3.981e-05	104527181	8	0.31
	Evaluation Method	False	85	1.817e-06	1.703e-05	871588819	4	0.14
		True	15	1.227e-06	1.091e-05	143537417	32	0.17
	Property	False	90	1.425e-05	2.934e-05	519800071	32	0.79
		True	21	1.209e-06	6.406e-06	475817201	8	0.65
SciBERT	Research Object	False	94	8.249e-05	2.969e-06	756272275	32	0.22
		True	10	1.001e-06	4.499e-05	357517963	32	0.19
	Evaluation Method	False	43	7.267e-06	1.144e-05	571598794	16	0.13
		True	38	8.367e-05	3.832e-06	177127938	8	0.14
	Property	False	91	1.865e-06	6.924e-06	899412442	16	0.76
		True	40	2.694e-05	7.450e-05	378601890	8	0.69
XLNet	Research Object	False	70	3.810e-05	1.269e-06	767864377	8	0.24
		True	40	3.635e-05	1.100e-06	1551604	8	0.33
	Evaluation Method	False	29	2.168e-05	3.345e-05	844604573	2	0.15
		True	44	6.961e-05	4.905e-06	532625637	8	0.26
	Property	False	28	2.096e-05	1.460e-05	289249899	2	0.71
		True	53	2.592e-05	3.760e-05	700183672	2	0.65

A.3.2. Visualization of the Best Results for Hyperparameter Search

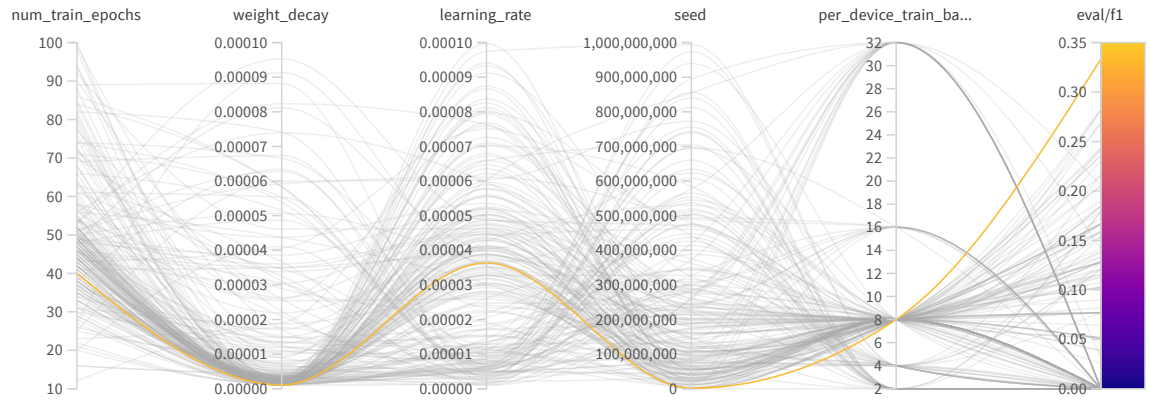

Figure A.2.: XLNet hyperparameter search for *Research Object* with the augmented dataset

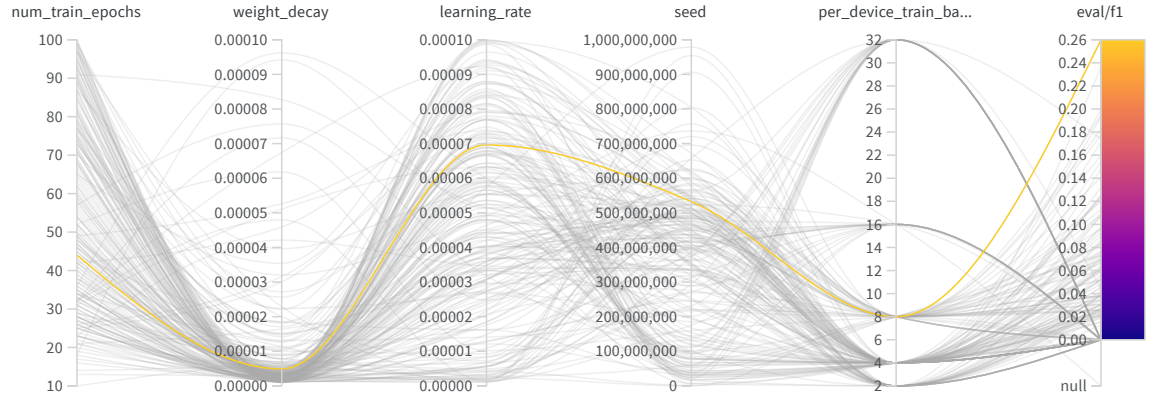

Figure A.3.: XLNet hyperparameter search for *Evaluation Method* with the augmented dataset

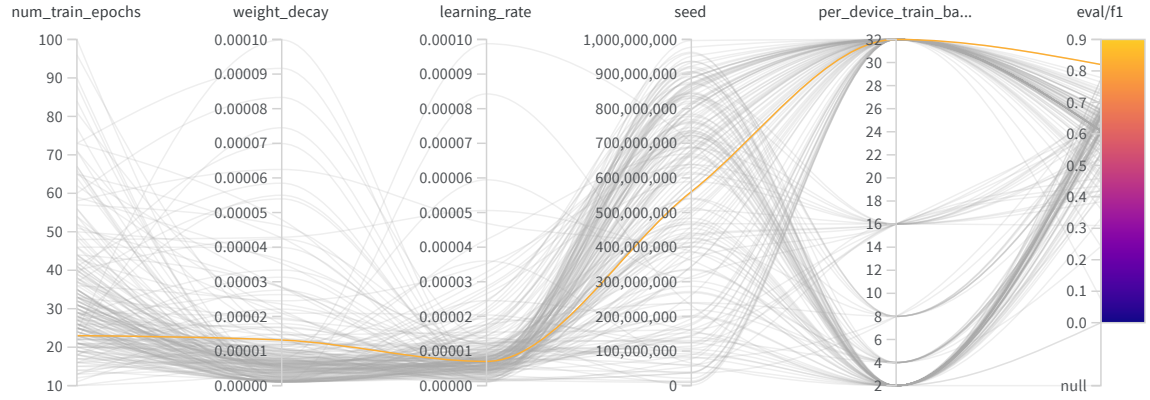

Figure A.4.: DeBERTa hyperparameter search for *Property* with the augmented dataset

A.3.3. Carbon Emission Results

Table A.3.: Carbon emission (kg) evaluation results for trained models using the basic dataset

Language Model	Run Type	Research Object	Evaluation Method	Property
BERT	hyperparameter search	0.220	**0.169**	**0.124**
	test	**0.002**	**0.003**	**0.002**
	author split	**0.002**	**0.003**	**0.002**
DeBERTa	hyperparameter search	*0.476*	*0.578*	0.288
	test	0.015	0.004	0.013
	author split	0.015	0.004	0.013
DistilBERT	hyperparameter search	0.221	0.428	*0.396*
	test	0.002	0.006	0.006
	author split	0.002	0.005	0.006
SciBERT	hyperparameter search	0.243	0.197	0.272
	test	0.010	0.004	0.009
	author split	0.010	0.004	0.009
XLNet	hyperparameter search	**0.217**	0.339	0.236
	test	0.010	0.005	0.004
	author split	0.010	0.005	0.005

Table A.4.: Carbon emission (kg) evaluation results for trained models using the augmented dataset

Language Model	Run Type	Research Object	Evaluation Method	Property
BERT	hyperparameter search	**0.251**	0.409	**0.245**
	test	0.001	0.007	0.002
	author split	0.002	0.006	0.002
DeBERTa	hyperparameter search	0.448	*1.613*	*0.438*
	test	0.005	0.015	0.005
	author split	0.005	0.013	0.005
DistilBERT	hyperparameter search	0.346	0.458	0.361
	test	0.007	**0.002**	0.002
	author split	0.008	**0.001**	**0.002**
SciBERT	hyperparameter search	0.318	**0.406**	0.344
	test	**0.001**	0.008	0.006
	author split	**0.001**	0.006	0.006
XLNet	hyperparameter search	*0.550*	0.535	0.350
	test	0.008	0.014	**0.013**
	author split	0.009	0.011	0.015

Table A.5.: Carbon emission (kg) evaluation results for the zero-shot classification approach

Language Model	Research Object	Evaluation Method	Property
GPT-3.5-turbo-0125	**0.004**	**0.004**	**0.004**
GPT-4-0125-preview	0.013	0.012	0.013
LLaMA2:7B	0.016	0.013	0.016
LLaMA213B	0.027	0.017	0.039
LLaMA2:70B	*0.234*	*0.155*	*0.322*
Mistral:7B	**0.009**	**0.009**	**0.011**
Mixtral:8x7B	0.024	0.024	0.026

A.4. Contribution C7: Additional Results for Retrieval

A.4.1. Evaluation Results for Question Complexity Types

Table A.6.: Evaluation results of the retrieval performance of the HubLink variants, considering the different question complexity types, based on graph variant **GV1**.

	Question Complexity	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	basic	**0.917**	**0.382**	**0.480**	**0.917**	**0.445**	0.490	**0.389**
	aggregation	0.810	0.209	0.285	0.497	0.225	0.347	0.240
	counting	0.840	0.275	0.372	0.644	0.357	0.526	0.340
	ranking	0.817	0.321	0.414	0.561	0.360	**0.576**	0.363
	comparative	0.742	0.262	0.366	0.456	0.320	0.560	0.296
	relationship	0.628	0.254	0.314	0.410	0.298	0.528	0.331
	negation	0.584	0.072	0.122	0.244	0.125	0.419	0.144
	superlative	0.656	0.129	0.193	0.319	0.207	0.540	0.237
HubLink (D)	basic	**0.861**	0.217	0.276	**0.611**	0.297	0.332	0.228
	aggregation	0.730	0.166	0.217	0.388	0.188	0.365	0.200
	counting	0.723	0.218	0.293	0.481	0.287	0.410	0.253
	ranking	0.659	0.221	0.278	0.428	0.278	0.494	0.269
	comparative	0.701	0.314	**0.376**	0.444	0.287	0.537	0.339
	relationship	0.689	**0.347**	**0.376**	0.456	**0.314**	0.627	**0.411**
	negation	0.639	0.065	0.118	0.325	0.169	0.534	0.200
	superlative	0.690	0.133	0.204	0.332	0.229	**0.635**	0.244
HubLink (F)	basic	**0.806**	0.279	0.338	**0.694**	**0.364**	0.392	0.287
	aggregation	0.652	0.215	0.277	0.427	0.172	0.267	0.235
	counting	0.779	0.273	0.376	0.477	0.301	0.561	0.273
	ranking	0.630	0.236	0.320	0.404	0.224	0.458	0.235
	comparative	0.617	0.366	0.428	0.504	0.355	0.549	0.366
	relationship	0.610	**0.420**	**0.443**	0.456	0.310	0.541	**0.428**
	negation	0.509	0.148	0.216	0.311	0.177	0.487	0.211
	superlative	0.575	0.251	0.315	0.325	0.232	**0.572**	0.263
HubLink (O)	basic	**0.806**	**0.280**	**0.345**	**0.806**	**0.590**	0.617	0.297
	aggregation	0.638	0.123	0.164	0.426	0.250	0.391	0.204
	counting	0.710	0.170	0.249	0.600	0.404	**0.714**	0.273
	ranking	0.611	0.128	0.188	0.394	0.256	0.514	0.234
	comparative	0.387	0.075	0.118	0.285	0.179	0.443	0.150
	relationship	0.465	0.265	0.274	0.328	0.259	0.653	**0.328**
	negation	0.499	0.039	0.067	0.232	0.121	0.475	0.144
	superlative	0.322	0.031	0.055	0.149	0.084	0.382	0.106

A.4.2. Evaluation Results for Use Cases

Table A.7.: Evaluation results of the retrieval performance of HubLink variants, considering the impact of different scholarly use cases.
The results are based on graph variant **GV1**.

	Use Case	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)	1	0.800	**0.507**	**0.575**	**0.767**	**0.557**	**0.644**	**0.552**
	2	**0.848**	0.252	0.364	0.729	0.301	0.341	0.281
	3	0.768	0.252	0.343	0.507	0.268	0.543	0.287
	4	0.663	0.198	0.277	0.395	0.266	0.561	0.255
	5	0.702	0.122	0.186	0.350	0.184	0.408	0.213
	6	0.779	0.206	0.286	0.428	0.278	0.512	0.257
HubLink (D)	1	**0.791**	**0.450**	**0.510**	**0.745**	**0.495**	0.556	**0.489**
	2	0.715	0.073	0.127	0.410	0.155	0.281	0.111
	3	0.675	0.265	0.332	0.444	0.234	0.459	0.298
	4	0.543	0.195	0.225	0.302	0.184	0.444	0.234
	5	0.756	0.144	0.191	0.317	0.183	0.481	0.236
	6	0.790	0.213	0.295	0.463	0.341	**0.691**	0.284
HubLink (F)	1	**0.770**	**0.524**	**0.591**	**0.758**	**0.499**	0.538	**0.533**
	2	0.674	0.150	0.228	0.438	0.202	0.272	0.158
	3	0.611	0.262	0.313	0.347	0.162	0.385	0.240
	4	0.524	0.268	0.330	0.390	0.254	0.503	0.291
	5	0.675	0.234	0.308	0.370	0.247	**0.597**	0.253
	6	0.676	0.271	0.336	0.489	0.293	0.534	0.306
HubLink (O)	1	0.689	**0.370**	**0.436**	0.667	**0.581**	**0.719**	**0.395**
	2	**0.776**	0.195	0.284	**0.690**	0.297	0.380	0.250
	3	0.531	0.152	0.162	0.374	0.275	0.512	0.254
	4	0.357	0.047	0.068	0.220	0.156	0.473	0.148
	5	0.463	0.044	0.077	0.251	0.150	0.484	0.135
	6	0.609	0.112	0.176	0.360	0.243	0.620	0.195

Table A.8.: Evaluation results of the retrieval performance for different scholarly use cases. The results are based on graph variant **GV1**.

	Use Case	Recall	Precision	F_1	Hits@10	Map@10	MRR@10	EM@10
HubLink (T)	1	0.800	**0.507**	**0.575**	**0.767**	**0.557**	**0.644**	**0.552**
	2	**0.848**	0.252	0.364	0.729	0.301	0.341	0.281
	3	0.768	0.252	0.343	0.507	0.268	0.543	0.287
	4	0.663	0.198	0.277	0.395	0.266	0.561	0.255
	5	0.702	0.122	0.186	0.350	0.184	0.408	0.213
	6	0.779	0.206	0.286	0.428	0.278	0.512	0.257
DiFaR	1	0.588	0.010	0.021	0.436	0.352	0.538	0.129
	2	0.104	0.001	0.001	0.000	0.000	0.000	0.000
	3	0.453	0.016	0.032	0.302	0.238	0.412	0.156
	4	0.264	0.011	0.021	0.191	0.145	0.324	0.121
	5	0.275	0.009	0.020	0.070	0.029	0.101	0.048
	6	0.421	0.014	0.031	0.255	0.152	0.400	0.155
Mindmap	1	0.177	0.034	0.053	0.005	0.001	0.006	0.004
	2	0.023	0.014	0.017	0.000	0.000	0.000	0.000
	3	0.208	0.033	0.054	0.035	0.004	0.007	0.007
	4	0.121	0.043	0.061	0.022	0.002	0.015	0.014
	5	0.098	0.034	0.048	0.016	0.005	0.041	0.012
	6	0.072	0.020	0.030	0.004	0.000	0.003	0.003
FiDeLiS	1	0.269	0.253	0.271	0.148	0.268	0.183	0.182
	2	0.125	0.043	0.065	0.036	0.125	0.055	0.025
	3	0.057	0.036	0.060	0.052	0.057	0.052	0.047
	4	0.061	0.036	0.150	0.053	0.061	0.056	0.043
	5	0.040	0.018	0.033	0.014	0.041	0.020	0.012
	6	0.047	0.025	0.076	0.030	0.047	0.035	0.031

A.4.3. Evaluation Results for Semi-Typed and Untyped Questions

Table A.9.: Evaluation results of the retrieval performance of HubLink variants and the baselines, considering the impact of semi-typed or untyped questions. The results are based on graph variant **GV1**.

Semi-Typed	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (T)							
True	**0.763**	**0.258**	**0.352**	**0.567**	**0.328**	**0.512**	**0.323**
False	0.747	0.233	0.302	0.456	0.268	0.488	0.273
HubLink (D)							
True	**0.730**	**0.229**	**0.302**	**0.461**	**0.272**	**0.493**	**0.284**
False	0.691	0.212	0.251	0.411	0.247	0.485	0.263
HubLink (F)							
True	**0.680**	**0.321**	**0.394**	**0.498**	**0.318**	**0.535**	**0.334**
False	0.622	0.237	0.296	0.407	0.218	0.415	0.248
HubLink (O)							
True	0.377	0.258	0.495	0.110	0.537	0.156	0.196
False	**0.441**	**0.287**	**0.565**	**0.179**	**0.583**	**0.222**	**0.251**
DiFaR							
True	0.256	0.189	0.349	0.011	0.359	0.023	0.128
False	0.155	0.109	0.238	0.010	0.343	0.021	0.078
MindMap							
True	0.006	0.001	0.006	0.029	0.105	0.042	0.005
False	0.024	0.004	0.021	0.031	0.133	0.047	0.010
FiDeLiS							
True	0.108	0.076	0.121	0.065	0.108	0.076	0.062
False	0.075	0.048	0.083	0.039	0.075	0.049	0.043

A.4.4. Evaluation Results for Hop Counts

Table A.10.: Evaluation results of the retrieval performance of HubLink (T) and the baselines, considering the impact of different hop counts.
The results are based on graph variant **GV1**.

	# Hops	Recall	Precision	F_1	Hits@10	MAP@10	MRR@10	EM@10
HubLink (F)	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	2	0.937	0.792	0.818	0.938	0.755	0.750	0.792
	3	0.732	0.358	0.446	0.699	0.427	0.509	0.404
	4	0.934	0.300	0.446	0.933	0.408	0.450	0.300
	5	0.813	0.212	0.313	0.524	0.268	0.459	0.246
	6	0.720	0.188	0.262	0.414	0.238	0.494	0.255
DiFaR	1	**1.000**	0.010	0.010	**1.000**	0.167	0.167	0.100
	2	0.667	0.007	0.013	0.333	0.153	0.250	0.050
	3	0.380	0.007	0.018	0.311	**0.307**	**0.469**	**0.113**
	4	0.300	0.004	0.004	0.000	0.000	0.000	0.000
	5	0.465	0.014	**0.028**	0.210	0.149	0.326	0.109
	6	0.285	**0.011**	0.022	0.176	0.120	0.262	0.108
MindMap	1	**1.000**	**0.070**	**0.130**	0.000	0.000	0.000	0.000
	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	3	0.136	0.031	0.048	0.005	0.001	0.006	0.004
	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.132	0.029	0.043	0.013	0.002	0.011	**0.009**
	6	0.115	0.033	0.048	**0.019**	**0.003**	**0.017**	0.008
FiDeLiS	1	**1.000**	**0.500**	**0.670**	**1.000**	**1.000**	**1.000**	**0.857**
	2	0.667	0.358	0.440	0.667	0.639	0.667	0.433
	3	0.060	0.037	0.045	0.060	0.052	0.062	0.037
	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.126	0.068	0.082	0.126	0.072	0.118	0.067
	6	0.050	0.031	0.035	0.050	0.021	0.070	0.024