Doctoral Thesis

An Investigation of Three Novel Applications for RDF-based Knowledge Graphs:

Unveiling Opportunities, Challenges, and Recommendations

Leon Martin

0000-0002-6747-5524

Media Informatics Group

University of Bamberg

Bamberg 2024

URN: urn:nbn:de:bvb:473-irb-1050589
DOI: https://doi.org/10.20378/irb-105058

Diese Arbeit hat der Fakultät WIAI der Otto-Friedrich-Universität Bamberg als Dissertation vorgelegen.

Gutachter: Prof. Dr. Andreas Henrich
Gutachter: Prof. Dr. Christoph Schlieder

Tag der mündlichen Prüfung: 15.11.2024

Contents

Danksagung (Acknowledgements) vii
English Summary ix
Deutsche Zusammenfassung (German Summary) xi

Introductory Paper 1
An Investigation of Three Novel Applications for RDF-based Knowledge Graphs: Unveiling Opportunities, Challenges, and Recommendations
1 Introduction 3
1.1 Research Questions 5
1.2 Thesis Structure 6
1.3 Software and Research Data 8
2 Research Context 10
2.1 Theoretical Background 10
2.2 Related Work on the Utilization of Knowledge Graphs 21
2.3 Scope and Size of Knowledge Graphs 25
3 Generation of Dual-Entity Knowledge Panels (Application 1) 27
3.1 Content and Positioning of **Paper I** 29
3.2 Content and Positioning of **Paper II** 32
3.3 Content and Positioning of **Paper III** 36
3.4 Presentation Formats for Entity Relationship Explanations 38
4 Import and Export of Research Contributions (Application 2) 41
4.1 Content and Positioning of **Paper IV** 42
4.2 Content and Positioning of **Papers V** and **VI** 44
5 Quality Assessment of Software Repositories (Application 3) 49
5.1 Content and Positioning of **Paper VII** 49
5.2 Content and Positioning of **Paper VIII** 53
6 Discussion of Insights 60
6.1 Opportunities 60
6.2 Challenges 65
6.3 Recommendations 71
6.4 Overview 75
7 Conclusion 77
7.1 Limitations and Future Work 77
List of Figures 80
List of Tables 82
List of Acronyms 83
Bibliography 84

Paper I 95
Fast Pathfinding in Knowledge Graphs Using Word Embeddings

Paper II 97
BiPaSs: Further Investigation of Fast Pathfinding in Wikidata

Paper III 99
A Testbed for Dual-Entity Knowledge Panels

Paper IV 101
On the Form of Research Publications for Use in Scientific Knowledge Graphs

Paper V 107
RDFtex: Knowledge Exchange Between LaTeX-Based Research Publications and Scientific Knowledge Graphs

Paper VI 109
RDFtex in-depth: knowledge exchange between LATEX-based research publications and Scientific Knowledge Graphs

Paper VII 111
Specification and Validation of Quality Criteria for Git Repositories using RDF and SHACL

Paper VIII 113
Assessing the FAIRness of Software Repositories using RDF and SHACL

Danksagung (Acknowledgements)

Nach über 20 Jahren der schulischen und akademischen Ausbildung stehe ich nun an einem Punkt, von dem ich zu verschiedenen Zeitpunkten nie geglaubt hätte, ihn erreichen zu können. Mit Demut blicke ich auf die vergangenen Jahre zurück und verspüre große Dankbarkeit für die Unterstützung, die mir zuteil wurde. Leider kann ich im Folgenden (auch aus Angst jemanden zu vergessen) nicht alle Personen nennen, die ich gerne nennen würde. Deswegen möchte ich zunächst einmal *allen* Danke sagen, die mich auf diesem Weg begleitet haben.

Ausdrücklich möchte ich mich jedoch bei Prof. Dr. Andreas Henrich bedanken, der durch seine Fachkompetenz und verständnisvolle Art einen Großteil zu meiner Entwicklung beigetragen hat. Auch den weiteren Mitgliedern meiner Promotionskomission, Prof. Dr. Daniela Nicklas und Prof. Dr. Christoph Schlieder, sowie allen Professorinnen und Professoren, die mich durch ihre Forschung und Lehre geprägt haben, möchte ich danken. Des weiteren möchte ich an dieser Stelle meine Mitdoktorandinnen und Mitdoktoranden Martin Bullin, Felix Engl, Tobias Gradl, Tobias Hirmer, Robin Jegan, Stefan Kufer, Michaela Ochs, Markus Wegmann und Adrian Wöltche nennen, die mir in zahlreichen Diskussionen wichtige Impulse für meine Forschung und Lehre gegeben haben. Einen wichtigen Beitrag haben auch die vielen Bacheloranden, Masteranden, Hiwis und Studierenden geleistet, die ich betreuen durfte. Leider ist die Liste zu lang, um Euch alle namentlich zu nennen. Seid Euch aber sicher, dass mir Eure Unterstützung sehr geholfen hat.

Nun möchte ich das Wort an diejenigen richten, die mir die Welt bedeuten. Dazu gehören zunächst die Mitglieder meiner Familie, insbesondere meine Frau Christina Kaiser, meine Mutter Andrea Martin, mein Bruder Lukas Martin und mein Stiefvater Matthias Tscherbner. Als nächstes folgen die Mitglieder der Crew, insbesondere Sascha Riechel, Tobias Schwartz, Jan Boockmann, Carlo Stingl und Melanie Vogel, sowie neben Robin Jegan und Felix Engl die weiteren Mitglieder des Cheese Platter Collective Andrea Papenmeier und Alexander Frummet. Zu guter Letzt sind noch meine alten Freunde Felix Hofmann und Matti Dorsch zu nennen. Es erfüllt mich mit großem Stolz, dass ich Euch alle in meinem Leben habe. Danke für alles!

Ich hoffe Euch ist bewusst, dass Ihr die folgenden knapp 80 Seiten sowie alle meine Forschungsartikel, die ebenfalls zu dieser Dissertation gehören, nun lesen müsst. Viel Spaß!

English Summary

Various sources suggest that Knowledge Graphs (KGs) despite being built upon decades-old foundations, are currently in the midst of a hype. The reason for this is rooted in the variety of applications, for which KGs can be used. While KGs have been employed successfully in many application domains such as information retrieval and recommender systems, recent research in machine learning and artificial intelligence highlights the opportunities that KGs provide in these domains. However, like any other hyped technology, KGs have been subject to inflated expectations, which has led a certain disillusionment in the recent years. To contribute to the further maturation of KGs towards a state with a deep and realistic understanding of the technology, this cumulative doctoral thesis investigates three novel applications of RDF-based KGs.

The first application leverages the large open-domain KG Wikidata to identify semantically useful paths between entities. These paths are then used as the basis for generating entity relationship explanations, which are to be displayed in the knowledge panels on the result pages of web search engines with the goal of satisfying the users' information need more quickly. The second application establishes a bidirectional knowledge exchange between (LaTeX-based) research publications and Scientific Knowledge Graphs (SKGs), one category of domain-specific KGs, through custom import and export commands. This technology is intended to narrow the gap between research publications and SKGs to facilitate the shift towards the envisaged KG augmented research. The third application leverages the reasoning capabilities of the RDF ecosystem to validate software repositories, represented as KG fragments, against quality criteria that they should meet according to the type of the respective project. In light of the still ongoing reproducibility crisis, the validation of software repositories against the FAIR principles is examined in particular.

Based on the insights obtained from the investigation of the three applications, opportunities, challenges, and recommendations for the utilization of KGs are derived. Some of the addressed aspects elaborate on topics discussed in previous work while others add novel points to this line of research. To give some examples, the aspects include remarks on the ability of KGs to integrate and interconnect heterogeneous data, the complexity of the RDF ecosystem, and data access facilities and interfaces.

Deutsche Zusammenfassung (German Summary)

Verschiedene Quellen deuten darauf hin, dass Knowledge Graphs (KGs; im Deutschen auch Wissensgraphen genannt), obwohl sie auf jahrzehntealten Grundlagen aufbauen, derzeit einen Hype erleben. Der Grund dafür ist die Vielfalt der Anwendungen, für die KGs genutzt werden können. Während KGs in vielen Anwendungsdomänen, wie z. B. Information Retrieval und Empfehlungssysteme, erfolgreich eingesetzt wurden, zeigt aktuelle Forschung im Bereich des maschinellen Lernens und der künstlichen Intelligenz neue Möglichkeiten des Einsatzes von KGs in diesen Bereichen auf. Wie bei jeder anderen Hype-Technologie wurden jedoch auch bei KGs überhöhte Erwartungen geweckt, was in den letzten Jahren zu einer gewissen Ernüchterung geführt hat. Um zur weiteren Reifung von KGs hin zu einem Zustand mit einem tiefen und realistischen Verständnis der Technologie beizutragen, untersucht diese kumulative Dissertation drei neuartige Anwendungen von RDF-basierten KGs.

Die erste Anwendung nutzt den großen Open-Domain KG Wikidata, um semantisch bedeutungsvolle Pfade zwischen Entitäten zu identifizieren. Diese Pfade dienen dann als Grundlage für die Generierung von Erklärungen der Beziehung zwischen den Entitäten, die in den Knowledge Panels auf den Ergebnisseiten von Web-Suchmaschinen angezeigt werden sollen, um den Informationsbedarf der Nutzer schneller zu erfüllen. Die zweite Anwendung etabliert einen bidirektionalen Wissensaustausch zwischen (LaTeX-basierten) Forschungspublikationen und Scientific Knowledge Graphs (SKGs), einer Kategorie von domänenspezifischen KGs, durch neuartige Import- und Exportbefehle. Diese Technologie soll die Lücke zwischen Forschungspublikationen und SKGs verkleinern, um den Übergang zu der angestrebten KG-gestützten Forschung zu erleichtern. Die dritte Anwendung nutzt die logischen Fähigkeiten des RDF-Ökosystems zur Validierung von Software-Repositories, die als KG-Fragmente dargestellt werden, gegen Qualitätskriterien, die sie je nach Art des jeweiligen Projekts erfüllen sollten. Angesichts der noch immer andauernden Reproduzierbarkeitskrise wird insbesondere die Validierung von Software-Repositorien gegen die FAIR-Prinzipien untersucht.

Auf Basis der Erkenntnisse, die bei der Untersuchung der drei Anwendungen gewonnen wurden, werden Möglichkeiten, Herausforderungen und Empfehlungen für den Einsatz von KGs abgeleitet. Einige der angesprochenen Aspekte vertiefen Themen, die bereits in früheren Arbeiten diskutiert wurden, während andere neue Aspekte beitragen. Angesprochen werden beispielsweise die Fähigkeit von KGs, heterogene Daten zu integrieren und miteinander zu verbinden, die Komplexität des RDF-Ökosystems sowie Datenzugriffsmöglichkeiten und Schnittstellen.

Introductory Paper

An Investigation of Three Novel Applications for RDF-based Knowledge Graphs:

Unveiling Opportunities, Challenges, and Recommendations

Leon Martin

1 Introduction

In an era characterized by the exponential growth of heterogeneous data, Knowledge Graphs (KGs) (Hogan et al., 2021) have emerged as a valuable technology. Despite being built upon decades-old foundations, their ability to offer a structured, interconnected representation of data addresses the pressing need for data organization and contextual understanding in today's information landscape. The surge in relevance of KGs is closely tied to the proliferation of diverse data sources and the continuous specialization of application domains. With information in the form of unstructured data, structured data, and everything in between, facilitating the creation of unified and coherent knowledge bases is paramount. For this purpose, KGs leverage semantic web standards, graph theory, and ontologies to map the intricate relationships between data points, fostering a unique contextual understanding of the data when applied correctly.

KGs have already been employed for a variety of tasks in numerous application domains including search engines, social networks, e-commerce, business intelligence, finance, and many more (Ji et al., 2022; Li et al., 2024; Zou, 2020). Nevertheless, the statistics of search engines and research platforms still report a continuously increasing interest in KGs today. For example, the Google Trends tool reports that between 2013 and 2023 the interest in KGs has doubled[1]. dblp's record even exhibits a far more drastic increase, listing 41 research publications on KGs that have been published in 2013 and 2,470 that have been published in 2023, as shown in Figure 1. A similar hype can also be observed for other technologies like blockchain (Belotti et al., 2019) and Large Language Models (LLMs) (Zhao et al., 2023). When it comes to the life cycle of hyped technologies, one can often intuitively distinguish three phases: In the first phase, there is a widespread application of the technology to diverse use cases, driven by the enthusiasm surrounding its potential. This phase usually results in a proliferation of applications, ranging from innovative and impactful to speculative and less effective.

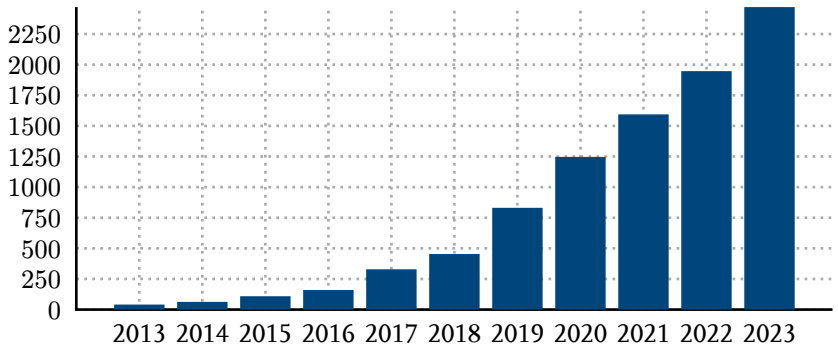


Figure 1: The number of publications returned in response to the query *knowledge graph* in dblp's record from 2013 to 2023. Data retrieved from https://dblp.uni-trier.de/search?q=knowledge+graph on 2024-07-03.

[1] https://trends.google.com/trends/explore?date=2013-10-19%202023-10-19&q=knowledge%20graph (visited 2024-07-11)

In the subsequent phase, a deliberate effort is made to gather and synthesize insights gained from the diverse applications of the technology. By systematically consolidating experiences and outcomes, a clearer understanding emerges. This enables the identification of common patterns, the discernment of opportunities, and the recognition of recurring challenges. Through consolidation, the noise created by the initial hype is filtered, allowing for a more nuanced and comprehensive assessment of the technology's real-world implications.

The resulting consolidated perspective serves as the basis for the third phase that involves drawing actionable conclusions. In this stage, recommendations for future applications are formulated based on the identified opportunities and challenges. This ensures that decisions regarding the technology's trajectory are well-informed, grounded in a comprehensive understanding of its impact, and positioned for effective applications. Figure 1 suggests that KGs are still in the middle of this process[2].

Other sources agree with this intuitive observation: Since KGs are a means of knowledge representation and thus relevant for Aritificial Intelligence (AI) and Machine Learning (ML), they are also featured in Gartner's Hype Cycle for Artificial Intelligence (Dedehayir & Steinert, 2016), an annual report that aims to identify AI innovations and techniques that offer significant and potentially transformational benefits. Figure 2 shows the positioning of KGs in the Hype Cycle over the last years. In the 2021 edition of the report (Goasduff, 2021), KGs almost reached the maximum of the entire process in a phase called *Peak of Inflated Expectations*, meaning that both industry and research are in a state of praising the possibilities the technology could potentially provide, in many cases without sufficient evidence. In the 2022 edition of the report (Wiles, 2022), KGs were still traveling on the *Peak of Inflated Expecta-*

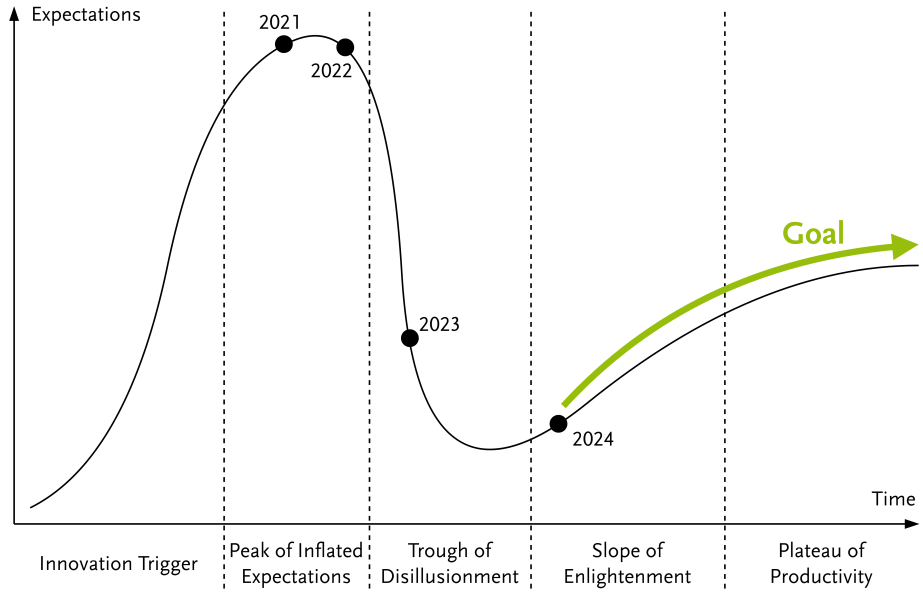


Figure 2: The positioning of KGs in the 2021, 2022, 2023, and 2024 editions of Gartner's Hype Cycle for Artificial Intelligence. Based on Goasduff (2021); Jaffri & Khandabattu (2024); Perri (2023); Wiles (2022).

[2] On the contrary, Google Trends and dblp's record both suggest that the hype for blockchain is calming down.

tions but passed the maximum. In the 2023 edition of the report (Perri, 2023), KGs progressed to the *Trough of Disillusionment*, the phase in which the initial inflated expectations meet the actual applicability of the technology, often with discouraging results. Finally, in the 2024 edition of the report (Jaffri & Khandabattu, 2024), KGs managed to escape the *Trough of Disillusionment* and entered the *Slope of Enlightenment*. To reach the desired final phase called *Plateau of Productivity*, the technology has to cross the *Slope of Enlightenment*.

From an organization's perspective, the adoption of a new technology is always accompanied by a certain amount of risk. Especially early adopters are prone to encountering unforeseen challenges regarding the utilization of a new technology in practice. When a technology reaches the *Plateau of Productivity*, a profound and realistic understanding of the opportunities, challenges, and recommendations associated with the utilization of the technology have emerged, facilitating a productive use of the technology with little risk. The 2024 Hype Cycle edition expects KGs to reach this phase in two to five years. The goal of this doctoral thesis is to contribute to the advancement of KGs as a technology by deriving opportunities, challenges, and recommendations for their utilization from an investigation of novel and diverse applications. The insights obtained this way complement remarks from previous surveys and other meta-level studies but also add new aspects that have not been addressed so far.

1.1 Research Questions

The utilization of KGs is a complex topic. As a result, it is not possible to provide a definite answer to all of its aspects in a single thesis. Instead, the goal of this thesis is to contribute to the understanding of the utilization of KGs based on an investigation of novel and diverse applications. Furthermore, this thesis mainly focuses on KGs based on the Resource Description Framework (RDF) (Cyganiak et al., 2014), one of the three popular options for implementing KGs[3]. Hence, this thesis aims to answer the following top-level research questions[4]:

Based on an investigation of novel and diverse applications,

1. what opportunities arise from the utilization of RDF-based KGs?
2. what challenges have to be overcome to utilize RDF-based KGs effectively?
3. what practices can be recommended for the effective utilization of RDF-based KGs?

The applications examined herein thus constitute the pillars of this thesis, serving as the foundational elements from which the overarching results are derived. Consequently, the presented insights are also limited to the selected applications, though. It is therefore particularly important to select the applications carefully in order to achieve the best possible outcome despite this limitation. Thus, the following three requirements have been established for the application selection:

[3] Section 2 elaborates on the reasons for this restriction.

[4] In this introductory paper, black rounded rectangles with arabic numbers indicate top-level research questions. The three top-level research questions complement the research questions that have been addressed in the supplementary papers that are part of this cumulative doctoral thesis.

1. The applications shall leverage KGs of different scopes and sizes.
2. The applications shall employ KGs for different purposes.
3. The applications shall be novel, i.e., address problems that are actively being researched or demonstrate new KG-based solutions for existing problems.

These requirements ensure the selection of applications that yield new and insightful perspectives on opportunities, challenges, and recommendations for the utilization of KGs. In detail, the first requirement leads to the selection of applications involving KGs with different vocabularies for encoding the knowledge relevant to their respective domains. Due to the correlation between the scope and size of KGs (cf. Section 2), this requirement also serves another goal, i.e., the selection of applications that pose algorithmic challenges due to the size of the employed KGs. While graph algorithms with a high time complexity are applicable for KGs with only a dozen data points, for example, they are impractical for open-domain KGs with millions of them, especially in time critical applications.

The intention of the second requirement is to guarantee that the results of this thesis are not limited to a narrow perspective on the utilization of KGs. Without this requirement, there would, for example, be a risk of overlooking challenges associated with integrating data into KGs if all applications solely focused on querying data from KGs. Additionally, major parts of the KG ecosystem would be left unused, thereby missing out on important practical aspects regarding the utilization of KGs. Again, it is impossible to provide a complete picture of this complex topic in a single thesis. Nevertheless, the applications have been selected carefully to cover important aspects including querying KGs, performing expensive algorithms on KGs, integrating data into KGs, and reasoning on KGs.

Finally, the third requirement acts as a safeguard to avoid that this thesis simply repeats previous research. As shown in Figure 1, the number of scientific articles mentioning KG published each year has drastically increased over the last decade. Thus, many topics have already been sufficiently covered. To avoid redundant research and obtain new insights regarding the opportunities, challenges, and recommendations for the utilization of KGs, it is mandatory to investigate novel topics that are subject of current research or to provide new KG-based solutions to existing problems.

Table 1 gives an overview of the applications covered in this thesis and their differentiating characteristics with respect to the set requirements. In total, three applications have been selected. That being said, ideas for several other applications have also been considered. However, they have been disregarded due to various reasons including lack of practical relevance and missing required resources.

1.2 Thesis Structure

As agreed with my doctoral committee, this doctoral thesis has the form of a cumulative thesis. It consists of this introductory paper and the supplementary papers listed in the back. Based on the insights obtained from the investigation of the applications, opportunities, challenges, and recommendations for the utilization of KGs are derived, which takes place in this introductory paper. Figure 3 illustrates this relationship and reveals, to which parts of the thesis the supplementary papers contribute.

Table 1: An overview of the three selected applications denoting the title of the applications as well as the scope(s) of and task(s) performed on the respectively employed KGs. The shown scope terminology will be further explained in Section 2.

	Application	Scope(s)	Task(s)
1	Generation of Dual-Entity Knowledge Panels	Open-Domain	Data Retrieval, Pathfinding
2	Import and Export of Research Contributions	Domain-Specific, KG Fragments	Data Retrieval, Data Integration, KG Construction
3	Quality Assessment of Software Repositories	KG Fragments	Reasoning, KG Construction

In total, this introductory paper and eight supplementary papers constitute this thesis. I am the first author in all of them except one, there is one publication that was written by me in sole authorship, and a part of them has been published in highly regarded outlets. Furthermore, all papers have been peer-reviewed. Thereby, the included papers fulfill the specific requirements set by my doctoral committee. Within this introductory paper, white rounded rectangles with roman numerals are used to refer to the individual papers, e.g., VII refers to the seventh supplementary paper. Although being accepted for publication, paper IV has not yet been published. Hence, a preprint version of this paper is attached in the back of the thesis, as required by the doctoral degree regulations of the University of Bamberg.

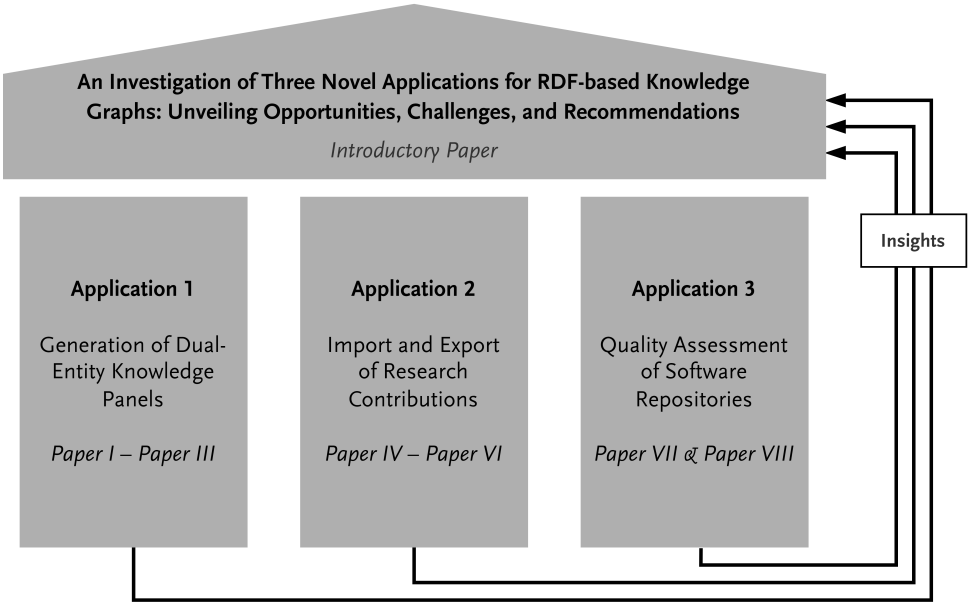


Figure 3: The structure of this doctoral thesis. The arrows illustrate that the insights from the investigation of the selected applications serve as the foundation for deriving opportunities, challenges, and recommendations for the utilization of KGs in this introductory paper.

While working on this doctoral thesis, I also supervised several bachelor theses, master theses, and student projects with valuable insights regarding the examined research topic. These works are referenced in the respective sections.

The remainder of this introductory paper is structured as follows: Section 2 focuses on the context of the presented research. This includes an introduction to essential components of the RDF ecosystem as well as a discussion of related work on the utilization of KGs. The next three sections, i.e., Section 3, Section 4, and Section 5, thoroughly examine the selected applications. Each application section comprises not only an introduction to the problem and a summary of the associated supplementary papers but, in parts, also additional remarks addressing remaining open questions. To keep this introductory paper concise, application-specific related work, examples, and discussions that are not particularly relevant for this thesis had to be omitted in the summaries. Hence, refer to the original papers for more details. Afterwards, representing the main contribution of this introductory paper, Section 6 derives opportunities, challenges, and recommendations for the utilization of RDF-based KGs with respect to the insights from Section 3, Section 4, and Section 5, thus answering the top-level research questions 1 , 2 , and 3 . The remarks in this section also take into account that some of the results apply to KGs in general and not only to KGs based on RDF. Finally, Section 7 concludes this introductory paper with a summary and a discussion of limitations and leads on future work.

1.3 Software and Research Data

Considering the still ongoing reproducibility crisis (Hutson, 2018), i.e., the fact that the results of a large number of research publications cannot be reproduced due to unavailable source code, it is important to me to point out the following: All software that has been implemented as part of this thesis and the associated research data is open-source and readily available on dedicated GitHub[5] repositories. To ensure reproducibility of results and portability, every program is set up for container virtualization via Docker[6] and accompanied by usage documentation. For persistence and future-proofing, the repositories are also indexed in the Software Heritage Project's archive[7]. Table 2 gives an overview of the applications and the corresponding repositories.

At this point, I also want to thank the creators and developers of Typst[8] and draw.io[9]. Without their tools, the process of authoring this thesis would have been significantly more challenging.

[5] https://github.com (visited 2024-07-11)
[6] https://www.docker.com (visited 2024-07-11)
[7] https://archive.softwareheritage.org (visited 2024-07-11)
[8] https://typst.app (visited 2024-07-29)
[9] https://www.drawio.com (visited 2024-07-29)

Table 2: An overview of the three examined applications and the corresponding GitHub repositories that comprise the implemented software and the associated research data.

Application		**Repositories**
1	Generation of Dual-Entity Knowledge Panels	https://github.com/uniba-mi/bipass-wikidata-pathfinder, https://github.com/uniba-mi/dual-entity-panels
2	Import and Export of Research Contributions	https://github.com/uniba-mi/rdftex
3	Quality Assessment of Software Repositories	https://github.com/uniba-mi/quare

2 Research Context

The first part of this section introduces the terminology, technologies, and standards used throughout this thesis. The second part discusses related work on the utilization of KGs. For conciseness, the discussion of some application-specific foundations and related work is postponed to the respective sections covering these applications.

2.1 Theoretical Background

Over time, different disciplines attempted to approach the abstract concept of knowledge in a formal way. For example, philosophy and psychology introduced categorizations that differentiate between categories of knowledge such as tacit knowledge, which includes mental models, and pragmatic knowledge, which encompasses the knowledge useful for an organization (Alavi & Leidner, 2001; De Jong & Ferguson-Hessler, 1996). As a precursor to KGs, semantic nets (Richens, 1956) have been proposed as an early technology for the formal representation of knowledge, facilitating reasoning when employed as a knowledge base. Even though the term has been coined several decades ago (Schneider, 1973), KGs only gained wide-spread attention in 2012 when Google introduced their KG as a technology to enhance their search engine (Singhal, 2012) through three capabilities:

1. Serving as a knowledge base facilitating the disambiguation of entities with similar labels.
2. Storing associated information on an entity, thereby allowing the generation of entity summaries that can be displayed in the form of knowledge panels.
3. Provision of links to related entities to support users in explorative search scenarios.

But, this purpose-centric description of a KG as a tool does not suffice as a definition of the concept. As a result, the term has been used in a variety of applications without a clear definition since then. Today, the term is typically not used to refer to the specific KG by Google but to denote the general idea of a KG, or more specifically, the different interpretations thereof. In fact, several competing definitions for the term emerged in the community. To resolve this issue, Ehrlinger & Wöß (2016) attempted to establish a standard definition based on an examination of similarities and differences of previously used definitions. Their definition goes as follows:

> "A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge."
>
> — Ehrlinger & Wöß (2016)

Notably, this generic definition does not impose any restrictions on the underlying data model and structure. Furthermore, it does not mention any standards or formal guidelines to be followed, when constructing a KG. However, the reason for this is not the lack of more formal definitions. For instance, Färber et al. (2018) define a KG in a more restrictive and technology-specific manner as[10]

[10] The meaning of the symbols and terminology mentioned in this definition is not important at this point.

> "[...] an RDF graph. An RDF graph consists of a set of RDF triples where each RDF triple (s, p, o) is an ordered set of following RDF terms: a subject $s \in U \cup B$, a predicate $p \in U$ and an object $U \cup B \cup L$. An RDF term is either a URI $u \in U$, a blank node $b \in B$, or a literal $l \in L$."
>
> — Färber et al. (2018)

Instead, the reason for the generic nature of their definition is that Ehrlinger & Wöß (2016) attempt to subsume the diverse previous interpretations of the term. At the same time, their definition still implies that a KG only qualifies as one if its purpose is to derive new knowledge using a reasoner, which is debatable. Even this thesis employs KGs that are not (primarily) intended to be processed with a reasoner. Accordingly, recent works state that the definition of a KG still remains contentious (Hogan et al., 2021). In contrast to the definition from Ehrlinger & Wöß (2016), the definition proposed by Hogan et al. (2021) focuses more on the knowledge to be encoded than the ultimate purpose:

> "[A knowledge graph is] a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities."
>
> — Hogan et al. (2021)

Since this definition aligns better with the topic of KG utilization, it is adopted for this introductory paper. In addition to the remarks on the history and the definition of KGs, Hogan et al. (2021) also provide detailed information on their implementation. Specifically, they present three data models to implement KGs, namely *directed edge-labelled graphs based on RDF* (Cyganiak et al., 2014), *property graphs* (Pokorný, 2015), and *heterogeneous graphs* (Wang et al., 2019). All of them are viable options but each comes with its own ecosystem. This thesis focuses on RDF-based KGs. The reason for this is related to the availability of necessary resources and tools. For instance, the application discussed in Section 3 requires a KG whose scope is comparable with the scope of web search engines like Google[11], Ecosia[12], Bing[13], and Startpage[14]. The only available high-quality KGs of this size are RDF-based, making the decision obsolete. That being said, the solutions proposed in this thesis could be adapted to KGs based on other data models given the necessary resources and tools, as well.

2.1.1 The Resource Description Framework

Motivated by Tim Berners-Lee's vision of the web's future, two powerful concepts emerged, namely the semantic web (Shadbolt et al., 2006) and linked (open) data (Bizer et al., 2023), that aim to extend the World Wide Web consisting of interlinked documents to a web of interlinked and machine-readable data, thereby unlocking the potential of the vast amounts of globally available information[15]. Paving the way for these concepts, standards including RDF and many related technologies surfaced, forming

[11] https://www.google.com (visited 2024-11-26)
[12] https://www.ecosia.org (visited 2024-11-26)
[13] https://www.bing.com (visited 2024-11-26)
[14] https://www.startpage.com (visited 2024-11-26)
[15] The interested reader is kindly referred to (Hogan, 2020) for a comprehensive introduction to the semantic web and linked (open) data.

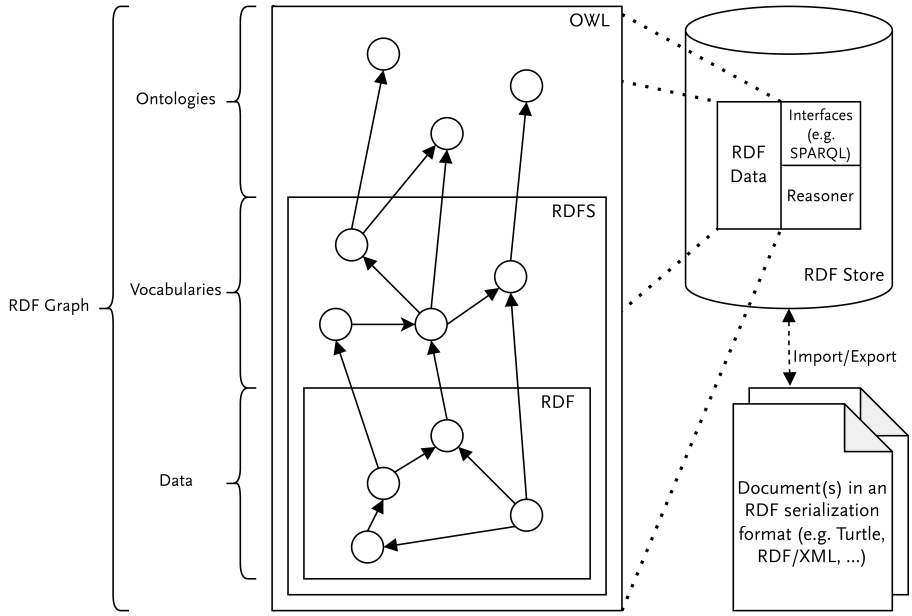


Figure 4: The parts of the RDF ecosystem relevant for this thesis. The depicted model is related to the variations of the semantic web stack (Horrocks et al., 2005) with an additional focus on the relationship between the technologies.

a large ecosystem. Over time, the ecosystem evolved through the introduction of new technologies that, in parts, also replaced previous approaches. Figure 4 provides an overview of the components of the RDF ecosystem that are relevant for this thesis and will thus be introduced in the following.

Fundamentally, RDF provides a generic approach for expressing information about arbitrary resources as triples (Cyganiak et al., 2014). To identify these resources, also called entities (Hogan et al., 2021), in the KG, Internationalized Resource Identifiers (IRIs) are employed, which are a superset of Uniform Resource Identifiers (URIs) with an extended set of permitted characters and therefore also a superset of Uniform Resource Locators (URLs) (Dürst & Suignard, 2005). Each triple consists of one *subject*,

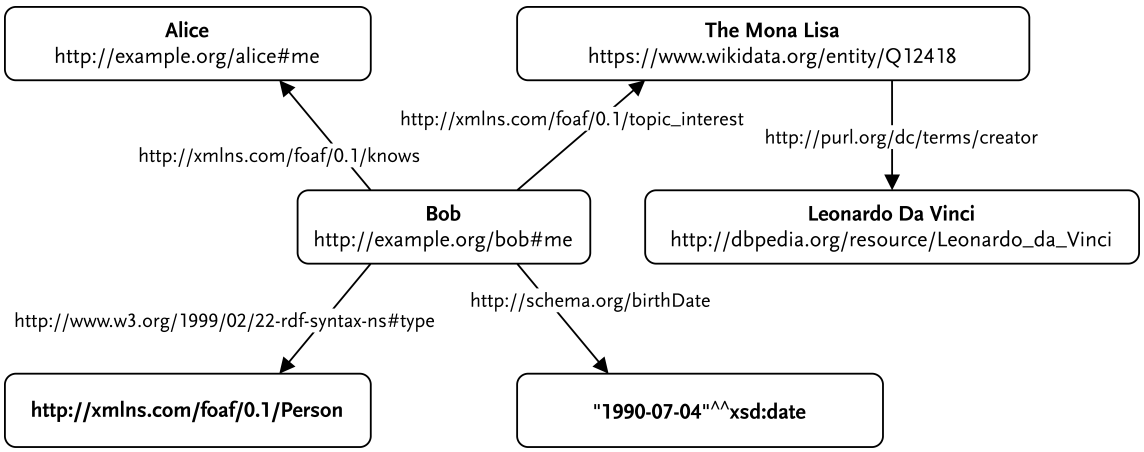


Figure 5: A simplified and slightly adapted version of the introductory RDF graph shown in the RDF documentation (Schreiber & Raimond, 2014).

which can be an IRI or a blank node, one *predicate*, which is an IRI, and one *object*, which can be an IRI, a literal, or a blank node. The predicate represents an instance of a property, i.e., a directed binary relation between the subject and the object. A property encodes some semantic statement that holds for the involved subject and object. By themselves, collections of such triples span graphs which are referred to as RDF graphs (Cyganiak et al., 2014). To give an example, Figure 5 shows an RDF graph consisting of five triples. For instance, there is one node representing the entity *The Mona Lisa* with the IRI https://www.wikidata.org/wiki/Q12418 and another one representing the entity *Leonardo Da Vinci* with the IRI https://dbpedia.org/resource/Leonardo_da_Vinci. In accordance with the definition by Hogan et al. (2021), the term entity not only covers single individuals but also classes of individuals and abstract concepts. Therefore, not only a software documentation but also *person* as a concept and emotions such as *love* qualify as entities, among others. As long as the nodes and edges of RDF graphs represent real-world entities and their relationships, they can be referred to as KGs[16]. Note that it is not mandatory to provide human-readable, i.e., natural-language, labels for the nodes of a KG, though.

In addition to entities, which are identified via IRIs, there are also literal nodes. To give an example, Figure 5 features a node with a literal, i.e., the string `"1990-07-04"`, representing a date in standardized XMLSchema notation (Malhotra et al., 2012). The nodes of the RDF graph are connected via distinct edges. They include an edge with the property http://purl.org/dc/terms/creator (the predicate) leaving the entity *The Mona Lisa* (the subject) and pointing to the entity *Leonardo Da Vinci* (the object). This triple denotes that *Leonardo Da Vinci* is the creator of *The Mona Lisa*. Although the properties are directed, they can usually be interpreted in both directions from a human point of view (Kasneci et al., 2009). Here, one could also state that *The Mona Lisa* has been created by *Leonardo Da Vinci*, effectively inverting the direction of the original property. This can be exploited for some applications including Application 1, discussed in Section 3.

Unlike property graphs, RDF does not allow providing additional information directly in nodes and edges. This raises the question how the labels depicted within the nodes of Figure 5 are actually stored. Like all pieces of information in RDF, such information has to be explicitly encoded using triples, as well. In the case of labels, certain vocabularies provide special properties to connect entities identified via an IRI with literals representing a human-readable label for them. One popular option in this regard is the `label` property from the `rdfs` vocabulary (Brickley et al., 2014). Accordingly, Figure 6 shows a more accurate representation of the RDF graph from Figure 5. The important role such vocabularies play in the RDF ecosystem is discussed in the following section.

Disregarding their concrete implementation, KGs can be interpreted as graphs in the formal sense to describe them using standard mathematical notation (Bollobás, 2002; Cormen et al., 2009). Hence, one can view a KG as a directed graph $G = (V, E)$, where the set of vertices/nodes V is the union of all subjects and objects present in the KG, and the set of edges E comprises all instances of properties present in the KG. The order of G is the number of nodes $|V|$ and the size of G is the number of edges $|E|$. Furthermore, the out-degree of a node equals the number of edges leaving it and the in-degree the number of edges pointing to it. The degree of a node is the sum of the in- and the out-degree.

[16] Section 6 elaborates on the relationship between RDF graphs and KGs, which can affect the feasibility of applications.

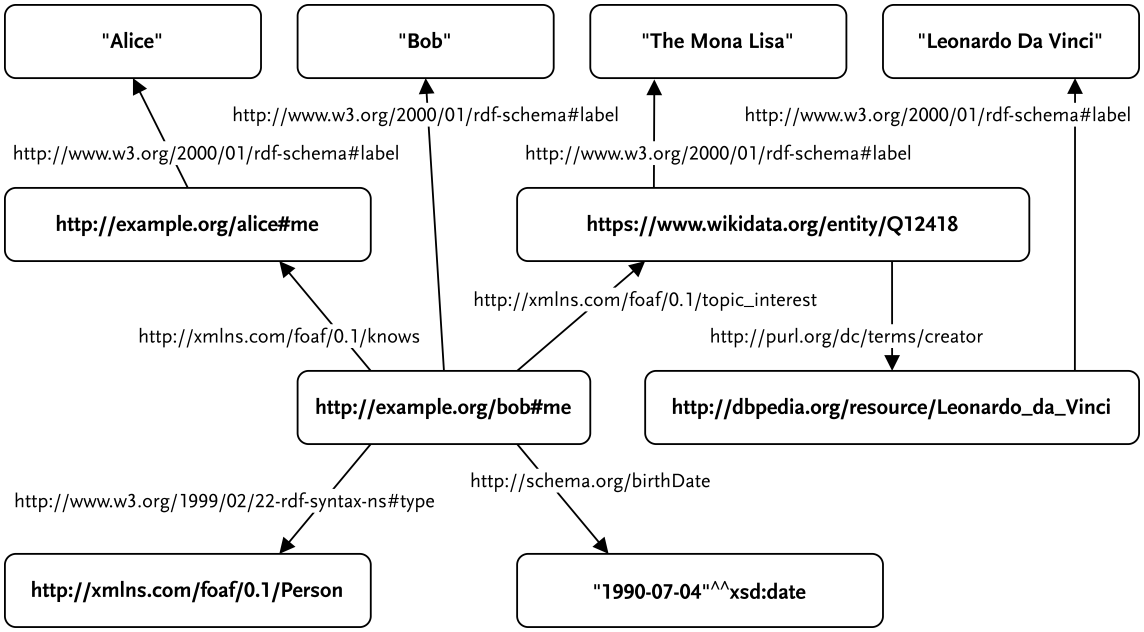


Figure 6: A more accurate visualization of the RDF graph from Figure 5 with additional triples encoding the entity labels explicitly.

2.1.2 Vocabularies and Ontologies

Fundamentally, IRIs are mere strings whose characteristics permit their usage as identifiers. As shown above, RDF offers a means to articulate statements about resources identified via IRIs by employing properties and values, without making assumptions about the semantic meaning of these IRIs (Schreiber & Raimond, 2014). However, communities utilizing RDF also require the capacity to define the semantic meaning of the terms they plan to employ in such statements. In the previous section the remark was made that the property IRI http://purl.org/dc/terms/creator expresses that the associated object is the creator of the associated subject. It is only through the semantic meaning attributed to resources identified via IRIs that an RDF graph is able to encode the application-specific knowledge (Brickley et al., 2014) the user communities are interested in.

In the RDF ecosystem, not RDF itself but its semantic extension called RDF Schema (RDFS) (Brickley et al., 2014) allows defining the semantic meaning of resources. Distinct collections of resources described via RDFS constitute so called RDF vocabularies (Schreiber & Raimond, 2014). Depending on the context, they are also called dictionaries (Brickley & Miller, 2014) or schemas (Guha et al., 2016) and sometimes even ontologies (Brickley & Miller, 2014; Hogan et al., 2021; Miles & Bechhofer, 2009). While the former two terms are unproblematic, the latter one adds a whole new, yet important dimension. For this reason, more information on the relationship between ontologies and RDF vocabularies follows in the lower part of this section.

As illustrated in Figure 4, RDFS and the vocabularies described with this technology follow the triple data model of RDF, thus ensuring a seamless integration with the ecosystem. RDF vocabularies are

application-specific in the sense that they cover a certain scope of semantic meaning. The following list describes some of the most popular[17] RDF vocabularies and their purpose:

1. RDFS provides an eponymous vocabulary (Brickley et al., 2014) that supports the description of resources and thereby the specification of other vocabularies. Hence, this vocabulary can be viewed as the fundamental vocabulary in the RDF ecosystem. To describe other resources, it introduces a class system for associating resources with RDF classes of which they are instances.
2. The Friend of a Friend (FOAF) vocabulary (Brickley & Miller, 2014) specializes in describing relationships between people as well as between people and information. Specifically, it defines the necessary classes and properties for encoding social networks, i.e., networks of human collaboration, friendship, and association, representational networks that describe a simplified view of a cartoon universe in factual terms, and information networks for linking independently published descriptions of this inter-connected world.
3. In theory, one can encode several knowledge organization systems including thesauri, classification schemes, subject heading lists, taxonomies, terminologies, and glossaries in RDF due to the generic nature of its data model (Miles et al., 2005). The purpose of the Simple Knowledge Organization System (SKOS) vocabulary is to facilitate this task by providing the necessary classes and properties to express such systems in triples (Miles & Bechhofer, 2009).
4. The focus of the vocabulary maintained by the Dublin Core Metadata Initiative (DCMI) lies on the provision of metadata for physical and digital media (Dublin Core Metadata Initiative, 2023). For this purpose, the vocabulary provides terms to describe the characteristics of physical or digital objects including their author(s), license, format, and many more.
5. Compared to other vocabularies, the schema.org vocabulary (Guha et al., 2016) that was founded by Google, Microsoft, Yahoo, and Yandex and is maintained by the community is different in the sense that it covers a particularly wide range of topics. This deliberate design choice was made to enable users to resort to one single vocabulary for their application rather than mix and match numerous other vocabularies, which might overlap in terms of their scope.

This list covers only a small selection of RDF vocabularies. In fact, there are so many vocabularies that vertical search engines for finding suitable candidates emerged. Examples are prefix.cc[18] and the KnowDive search engine[19] that includes 849 linked open RDF vocabularies (Vandenbussche et al., 2017). Thus, the schema.org community's effort to provide one vocabulary covering a variety of topics is commendable as it makes the RDF ecosystem more accessible (Guha et al., 2016).

To employ a third-party vocabulary specified with RDFS in an application, one simply has to use the terms of the vocabulary in their RDF graph; no further configuration required. Usually, the terms of a vocabulary share a namespace, which is the former part of the terms' IRIs, in this context. For example, all classes and properties present in the SKOS vocabulary share the namespace http://www.w3.org/2004/02/skos/core#. Within this namespace, one can find the class http://www.w3.org/2004/

[17] Richard Cyganiak, one of the authors of Cyganiak et al. (2014) offers a ranking of popular RDF vocabularies at http://richard.cyganiak.de/blog/2011/02/top-100-most-popular-rdf-namespace-prefixes (visited 2024-07-30).
[18] https://prefix.cc (visited 2024-08-01)
[19] https://lov.linkeddata.es/dataset/lov (visited 2024-05-22)

02/skos/core#Collection but also the property http://www.w3.org/2004/02/skos/core#hasTopConcept. Occasionally, one vocabulary has different namespaces for classes and properties or for certain parts of the semantic scope. It has become established that practitioners take advantage of the fact that URLs are a subset of IRIs by using URLs as identifiers for entities and properties such that users have easy access to additional information. For instance, entering http://www.w3.org/2004/02/skos/core#Collection in a browser takes users to a documentation page, while entering https://www.wikidata.org/wiki/Q12418 takes users to the Wikidata page of *The Mona Lisa*.

From a reasoning point of view, RDF is an assertional language for expressing statements on resources. It is specified that each semantic extension on top of RDF, including RDFS, must not modify or negate the minimal truth conditions that come with RDF (Hayes & Patel-Schneider, 2014). Given these conditions, interpretations of RDF graphs can be made that include certain entailments. While RDF semantics only allow basic entailments to be made, the added expressive power of RDFS semantics facilitates entailments regarding the transitivity of the resources' class affiliations, for example. For applications that require more complex reasoning on RDF graphs, the RDF-based semantics variant of the Web Ontology Language (OWL) (Schneider et al., 2012) is a popular choice. As suggested by its name and depicted in Figure 4, it facilitates modelling ontologies and, being a semantic extension of RDF and RDFS, also relies on the triple data model.

The term ontology has been in use for a long time and has different meanings depending on the context. From a computer science perspective, an ontology is a computational artifact modelling the structure of a system with its relevant entities and their inter-relationships in a formal way[20] (Guarino et al., 2009; Uschold & Gruninger, 2009). Such relationships can have a variety of characteristics. Some relationships might be equivalent to others, some might be the opposite of others, some entity might only qualify as an instance of a certain class if it possesses certain relationships, and so on. A powerful knowledge representation language such as OWL can model these intricate characteristics (Hitzler et al., 2012). Since OWL is closely related to description logics (Baader et al., 2003), one can use semantic reasoners such as Pellet (Hitzler et al., 2012) to infer diverse entailments on OWL-enhanced RDF graphs. For Application 3 discussed in Section 5, this capability is leveraged.

2.1.3 Serialization Formats and Path Notation

For exchanging and archiving RDF data, various serialization formats exist (Schreiber & Raimond, 2014), including the following popular examples:

1. Turtle, a member of the Turtle family of RDF languages
2. RDFa, a syntax for embedding RDF data in HTML and other XML-based languages
3. JSON-LD, an RDF syntax based on JSON
4. RDF/XML, an RDF syntax based on XML

[20] Considering this generic description of ontologies, RDF vocabularies also qualify as ontologies as they define classes and properties to describe things, albeit in a much simpler form than OWL. This answers the question why vocabularies specified using RDFS are sometimes also called ontologies, as stated before.

```
BASE   <http://example.org/>
PREFIX rdfs: <http://w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <http://schema.org/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX dbpedia: <http://dbpedia.org/resource/>

<bob#me>
    rdfs:label "Bob" ;
    a foaf:Person ;
    foaf:knows <alice#me> ;
    schema:birthDate "1990-07-04"^^xsd:date ;
    foaf:topic_interest wd:Q12418 .

<alice#me> rdfs:label "Alice" .

wd:Q12418
    rdfs:label "Mona Lisa" ;
    dcterms:creator dbpedia:Leonardo_da_Vinci .

dbpedia:Leonardo_da_Vinci
    rdfs:label "Leonardo Da Vinci" .
```

Listing 1: The RDF data from Figure 6 in Turtle format

Disregarding the employed format, the RDF documents resulting from the serialization of a set of triples are semantically equivalent since they express exactly the same triples without any semantic changes but only in a different syntax (Schreiber & Raimond, 2014). For this thesis, the first two are the most relevant serialization formats. The Terse RDF Triple Language (Turtle) (Beckett et al., 2014) expresses RDF data in a rather human-readable and therefore useful syntax in terms of maintainability. Listing 1 shows the RDF data from Figure 6 serialized using the Turtle format. To avoid lengthy IRIs in the actual data, Turtle offers a shorthand notation for specifying an arbitrary number of prefixes that can be used to substitute parts of the full IRIs, typically the RDF vocabularies' namespaces. By specifying the `BASE` to be `<http://example.org/>`, one can address the entity referring to Bob using `<bob#me>` instead of `<http://example.org/bob#me>`. Further, using the `PREFIX` keyword, the prefix `dcterms` is specified as a substitute for `<http://purl.org/dc/terms/>` such that the property `<http://purl.org/dc/terms/creator>` can be written as `dcterms:creator`. Many other technologies in the RDF ecosystem feature similar means of abbreviating IRIs.

The RDF in Attributes (RDFa) format (Herman et al., 2015) allows the enrichment of HTML and actually any XML-based language with RDF data through special attributes conforming with the XML standard. To give an example, Listing 2 shows the RDF data from Figure 6 embedded in an HTML document using RDFa. Similar to Turtle, RDF provides means of specifying prefixes to abbreviate IRIs, in this case via the `prefix` attribute attached to the `body` element. One difference between the two formats is that RDF is able to use strings in the HTML document both as values of the HTML element and as

```
<body
  prefix="foaf: http://xmlns.com/foaf/0.1/
     schema: http://schema.org/
     dcterms: http://purl.org/dc/terms/
          rdfs: http://w3.org/2000/01/rdf-schema#"
>
  < div resource="http://example.org/bob#me" typeof="foaf:Person">
   <  p>
    <   span property="rdfs:label">Bob</span> knows
    <   a property="foaf:knows" href="http://example.org/alice#me">a person</a>
    and was born on the
    <   time property="schema:birthDate" datatype="xsd:date">1990-07-04</time>.
   </  p>
   <  p>
    Bob is interested in
    <   span
         property="foaf:topic_interest"
         resource="http://www.wikidata.org/entity/Q12418"
         >the Mona Lisa</span
    >.
   </  p>
  </ div>
  < div resource="http://example.org/alice#me">
   <  p>This person is called<span property="rdfs:label">Alice</span>.</p>
  </ div>
  < div resource="http://www.wikidata.org/entity/Q12418">
   <  p>
    <   span property="rdfs:label">The Mona Lisa</span> was painted by
    <   a
         property="dcterms:creator"
         href="http://dbpedia.org/resource/Leonardo_da_Vinci"
         >a famous painter</a
    >.
   </  p>
  </ div>
  < div resource="http://dbpedia.org/resource/Leonardo_da_Vinci">
   <  p>
       This painter is is known as<span property="rdfs:label"
         >Leonardo Da Vinci</span
    >.
   </  p>
  </ div>
</body>
```

Listing 2: The RDF data from Figure 6 embedded in the body of an HTML document.

the objects the RDF triples. For instance, the string `Leonardo Da Vinci` serves as the element value of the last `span` element in the document and is therefore displayed when the document is opened in browser. At the same time, the string serves as the object of the triple denoting the label for the entity referring to *Leonardo Da Vinci*. Note how RDFa exploits the tree structure and nesting of XML to asso-

ciate subject, predicate, and object of each triple. In summary, RDFa can thus be viewed as a means of adding RDF-compatible semantic markup to XML documents in XML-compliant syntax.

Later sections of this thesis unveil that RDF serialization formats are not suitable for end-users, though. Accordingly, the remainder of this introductory paper uses a custom notation to convey triples in a concise and human-readable format. In this notation, the triple from Figure 6 looks as follows:

The Mona Lisa —dcterms:creator→ Leonardo da Vinci

Using concatenation, this notation is also able to denote longer paths, even with properties alternating in terms of their direction. Again, using the same example, one can express a longer path like this:

"1990-07-04" ←schema:birthDate— Bob —foaf:topic_interest→ The Mona Lisa
—dcterms:creator→ Leonardo da Vinci

As shown, the notation follows the idea of omitting lengthy IRIs, as well. To this end, it provides the human-readable labels of triple components in cases where labels are available. Alternatively or if the KG lacks a label for a certain component, the shorthand notation is used after mentioning the relevant prefixes in the respective paragraphs. Occasionally, extra information like proprietary entity and property IDs from specific KGs is provided in the notation, as well.

2.1.4 RDF Stores and SPARQL

Looking for a technology to store and retrieve RDF data some time ago, one would have quickly encountered *triple stores* (Rusher, 2001), sometimes also spelt *triplestores* (Saleem et al., 2019). Today, this notion is not as common anymore. For example, the documentation of the Ontotext GraphDB[21], a what they call RDF database, mentions the fact that this technology is a triple store only in some older and not particularly prominent places. Amazon's option, called Amazon Neptune[22], is also not explicitly termed as a triple store, even though it is listed in the documentation of Eclipse RDF4J[23], formerly known as Apache Sesame, as a third-party triple store compatible with the RDF4J platform. Virtuoso[24], another example, is referred to as a combined multi-model database management system and virtualization platform featuring RDF data management and RDF middleware. The term triple store is mentioned nowhere in its documentation. The same holds for Blazegraph[25], one of the central technologies powering Wikidata, as well as the compact RDFLib[26] library, which is called a pure Python package for working with RDF. Hence, all types of RDF storing technologies are subsumed under the term RDF store in the remainder of this introductory paper.

For this thesis, the implementation details of RDF stores are not important. What is important, though, is that RDF stores support manipulative operations on the RDF data similar to the Create, Read, Up-

[21] https://www.ontotext.com/products/graphdb (visited 2024-07-11)
[22] https://aws.amazon.com/de/neptune (visited 2024-07-11)
[23] https://rdf4j.org (visited 2024-07-11)
[24] https://virtuoso.openlinksw.com (visited 2024-07-11)
[25] https://blazegraph.com (visited 2024-07-11)
[26] https://rdflib.readthedocs.io/en/stable/index.html (visited 2024-07-11)

```
# Dedicated to my friends from the Cheese Platter Collective
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?cheese ?cheeseLabel ?origin ?originLabel
WHERE
{
  ?cheese wdt:P31 wd:Q10943 .
  ?cheese wdt:P495 ?origin .

  ?cheese rdfs:label ?cheeseLabel .
  ?origin rdfs:label ?originLabel .

  FILTER(
    LANG(?cheeseLabel) = "en" &&
    LANG(?originLabel) = "en" &&
    STRSTARTS(STR(?cheeseLabel), "A")
  ) .
} ORDER BY ASC(?cheeseLabel)
```

Listing 3: A simple SPARQL query for retrieving certain types of cheese from Wikidata.

date, Delete (CRUD) operations known from other types of databases (Truica et al., 2015). To this end, they typically provide APIs with programmatic access to the raw RDF data. In addition, endpoints for issuing queries in the SPARQL Protocol And RDF Query Language (SPARQL) (W3C SPARQL Working Group, 2013) represent another alternative that is also often provided[27].

Listing 3 shows a simple SPARQL query that retrieves a selection of certain cheese types available in Wikidata when issued to the Wikidata Query Service[28]. Although SPARQL's syntax is clearly influenced by SQL, there are significant differences due to the variations in the underlying data models. In the first part of the exemplary query, the `PREFIX` keyword is used to define prefixes for abbreviating the namespaces of the employed vocabularies. Then, four variables are declared next to the `SELECT` keyword,

Table 3: The result returned by the Wikidata Query Service in response to the query from Listing 3. Data retrieved from https://w.wiki/8p2o on 2024-01-11.

cheese	cheeseLabel	origin	originLabel
http://www.wikidata.org/entity/Q12275182	Abbey of Westmalle	http://www.wikidata.org/entity/Q31	Belgium
http://www.wikidata.org/entity/Q120968865	Alta Badia	http://www.wikidata.org/entity/Q38	Italy
http://www.wikidata.org/entity/Q111835885	Argentine cheese	http://www.wikidata.org/entity/Q414	Argentina
http://www.wikidata.org/entity/Q6094551	Arribes de Salamanca Cheese	http://www.wikidata.org/entity/Q29	Spain

[27] The Neo4J graph database (https://neo4j.com/product/neo4j-graph-database; visited 2024-01-10) uses the query language Cypher for performing CRUD-like operations on KGs based on property graphs.
[28] https://query.wikidata.org (visited 2024-08-12)

which will represent the columns the result will comprise. The subsequent `WHERE` clause provides the graph patterns to be matched against the RDF graph. In this clause, filters can be applied via the `FILTER` keyword to narrow the results, as well. Each solution for this `WHERE` clause is listed as a distinct entry in the result. The results can also be modified, in this case they are sorted with respect to the label of the cheese type in ascending order using `ORDER BY ASC`. In summary, this query retrieves all cheese types in Wikidata that have an english label starting with the letter A and a known country of origin with an english label, sorted with respect to the label of the cheese type in ascending order. Table 3 shows the result of the query. Even this simple query, which showcases only a fraction of SPARQL's capabilities, already appears rather complex. Accordingly, only expert users can be expected to use SPARQL endpoints, as indicated in Figure 4. For end-user interfaces, another layer of abstraction is required.

Apart from the ability to perform manipulative operations, RDF stores usually also support the export of RDF data as RDF documents in a serialization format of choice as well as the import of RDF data from external RDF documents[29], as indicated in Figure 4. Furthermore, some RDF stores also facilitate reasoning in varying degrees. Ontotext GraphDB, for example, is capable of reasoning with respect to some variants of OWL, while RDFlib lacks reasoning capabilities, thus calling for additional reasoning technologies if an application demands it.

The most important aspects of the RDF ecosystem that are relevant for understanding the rest of this thesis have now been covered. Before proceeding to an examination of the three selected applications, though, the following section first investigates related work on the utilization of KGs.

2.2 Related Work on the Utilization of Knowledge Graphs

In light of the topic of this thesis, basically all endeavors that utilize KGs in some way can be considered related work. Anyhow, since KGs have been applied for a numerous tasks in a variety of application domains in the past, it is neither possible nor expedient to discuss all of them here. Hence, this section opens with an investigation of surveys and other meta-level studies on the topic of KGs utilization to provide an overview of this research area. The first survey (Zou, 2020) to be mentioned in this context distinguishes five application domains for KGs:

1. The *question answering* application domain encompasses all question answering systems that leverage the semantic information from KGs to enhance search results. The systems can be further categorized depending on how they parse and transform natural language questions, and how they identify and rank candidate answers. For instance, information retrieval based systems try to automatically translate natural language questions into structured queries that can be executed on KGs, whereas embedding based systems create vector representations of natural language questions and calculate their similarity to vector representations of data in KGs.

[29] Databases for property graphs like the Neo4J graph database (https://neo4j.com/product/neo4j-graph-database; visited 2024-01-10) also support the import and export of RDF documents and thereby facilitate switching between the RDF data model and the property graph data model underlying KGs.

2. In the domain of *recommender systems*, KGs can be used to increase the recommender accuracy and the diversity of recommended items. Here, embedding based systems calculate knowledge graph embeddings and supply the learned entity embeddings to a recommendation framework, while path based systems leverage the varied connection patterns in KGs to provide additional information for recommendations.
3. There are multiple ways of leveraging KGs in the domain of *information retrieval systems*. Queries can be expanded with entities related to mentioned entities and documents can be enriched by adding annotated entities to the vector space representations of documents. Furthermore, the connections in KGs can be exploited to improve the document ranking.
4. The *domain-specific* category of application domains features works that utilize KGs in the medical domain, the cybersecurity domain, the financial domain, the news domain, and the education domain. Compared to the other application domains, the domain-specific knowledge from the individual domains is pivotal for the application at hand.
5. The final application domain category accommodates further works that do not fit into the previous application domains. These include applications in the context of social networks, classification, and geoscience, among others.

By describing a variety of applications of KGs, this survey gives an idea of the opportunities that arise from the utilization of KGs. However, the work lacks a summary of these opportunities and also remains shallow in terms of challenges and recommendations regarding the utilization of KGs. Another survey (Ji et al., 2022) approaches the topic of KG utilization from a research-oriented perspective by distinguishing four categories of KG research:

1. Knowledge representation learning encompasses research on the distributed learning of embeddings that appropriately reflect the rich semantic information of entities and relation in KGs. This task includes research on the representation space, in which entities and relations are embedded, on the scoring function, which measures the plausibility of triples, on encoding models, which encode interactions of entities and relations into the representation space, and on auxiliary information, which is incorporated into the embedding methods in order to increase the systems' performance.
2. The goal of knowledge acquisition is to construct KGs from unstructured, semi-structured, and structured data, to fill in gaps in existing KGs, and to discover and recognize entities and relations. This translates to three principle task, namely KG completion, which focuses on predicting links, entities, and relations, entity discovery, which encompasses all sub-tasks that explore entity-related knowledge including entity recognition, entity disambiguation, entity typing, and entity alignment, and relation extraction, which attempts to automatically construct or complete KGs by extracting triples from plain text.
3. Research on temporal knowledge graphs explores approaches for injecting temporal information into KGs to be able to express that some facts are only valid in a certain time period and that knowledge evolves over time. This contrasts the typical static KGs, in which knowledge has no temporal limitations.

4. The fourth research category dives into what they call knowledge-aware applications. With the remark that the survey only covers some examples of them, they discuss applications in language representation learning, where attempts are made to integrate the symbolic factual knowledge from KGs into state-of-the-art language models, as well as in question answering and in recommender systems like the first survey (Zou, 2020) does.

By highlighting three concrete research problems before diving into knowledge-aware applications, this survey provides more information on the challenges that surface when employing KGs for downstream applications. For example, the challenge of incomplete information, which is tackled using KGs completion techniques, is pivotal in applications like question answering and recommender systems whose performance heavily depends on the availability of knowledge. That being said, the survey does not attempt to systematically summarize these challenges. The next work (Li et al., 2024) to be mentioned here, discusses the utilization of KGs from the users' point of view, based on an interview study conducted with 19 KG experts from federally funded research and development centers, academia, and enterprises. In this study, the interviewees have not only been interrogated regarding the KG tools they use and the use cases they focus on but also regarding the benefits the usage of KGs provides and possible challenges that occur in practice. With respect to benefits and affordances of using KGs, the authors distill three aspects from the responses:

1. Several interviewees highlight the schema flexibility of KGs as a central advantage of KGs, especially compared to traditional relational databases, as it allows adding new data just as required with little effort.
2. With respect to end-user applications, the interviewees commend the ability of KGs to integrate both public and non-public data from multiple data sources and domains in order to generate and contextualize their KGs. While individual associations usually do not provide value on their own, investigating large numbers of associations in this data facilitates the identification of novel relationships and clusters.
3. The combination of the semantic nature of KGs and its robust data management capabilities provide benefits for many use cases. For these benefits, it is justifiable to accept the additional effort caused by the complexity of the graph modeling and the use of additional query languages.

The reason why this list might give the impression of being rather superficial is that the clear focus of the study lies on the challenges of KG utilization and the discussion of KG visualization approaches that can mitigate these challenges. Hence, moving on to the challenges of the utilization of KGs, the study identifies four categories with several aspects in each of them (Li et al., 2024):

1. Data quality issues are the most common challenges faced by the interviewees. In particular, sparse or missing data, incorrect or unverifiable data, obsolete data, and duplicate entities are problematic. Most of these issues surface in incomplete or in-progress enterprise KGs, while public KGs such as Wikidata can be more complete. This difference by itself poses problems, as well. For example, the performance of ML models can vary significantly between Wikidata and enterprise KGs with missing data, posing the risk of inflated expectations. To maintain the quality of KGs, i.e., resolve the data quality issues listed above, manual updates, often by domain

experts, are required to curate the data, in many cases even if approaches for the automatic generation of KGs are employed.

The problem of duplicate entities, i.e., entities in KGs that actually refer to the same entity, can compromise the performance of applications that have to pinpoint specific entities in the data. Another entity related issue is that, in the context of pathfinding in KGs, densely connected nodes in between target nodes can grossly inflate the number of discovered paths, leading to irrelevant outputs. This problem in particular also affects Application 1 discussed in Section 3 because Wikidata also features entities with exceptionally high degrees, thus posing challenges regarding the pathfinding required for this application.

2. Querying data from KGs is central for many applications. As explained above, KGs can be implemented based on different data models which necessitates learning the corresponding query languages, e.g., SPARQL for RDF-based KGs. This cannot be expected from end-users, raising the need for more abstract user interfaces. But even for expert users, constructing queries in these different languages is cumbersome. Some interviewees also complain about the lack of interim search results, i.e., that many systems only provide search results once the entire result set is compiled, which can become especially frustrating when querying slow systems.
3. The first aspect in the category of socio-technical problems is related to the incomplete understanding of end-users' needs. In this regard, some interviewees criticize KG practicioners that begin creating new KGs or add unnecessary features before completely understanding the requirements resulting from the needs of the end-users. Since the creation of proper KG is a complex tasks, a lot of resources can get wasted this way.

   Another significant challenge is the lack of standardized nomenclature, where different groups might use the same word with multiple meanings or different terms for the same concept. Some interviewees stated that, for example, the concept of profit can be interpreted in various ways. When asking different individuals about it, each might provide a different explanation.

   Another issue are organizational politics and unsustainability. In enterprise settings, KGs often fail due to political reasons or long-term unsustainability. For instance, the success of a KG may be hindered by a lack of ongoing interest, support, and resources from leadership, leading to its eventual abandonment. Organizational issues are frequently cited as major reasons why AI and ML models fail to be adopted in industry settings.

   Moreover, while research continues to focus on optimizing KG databases and query languages, numerous technological problems have already been resolved. The primary challenge lies in addressing the social issues related to KGs. Therefore, some interviewees stated that the focus should shift from developing new tools to effectively utilizing existing ones. This underscores the need for computer scientists to engage more with these social challenges.
4. In the final category, the authors focus on the flaws of node-link diagrams, the most common approach for visualizing KGs. According to the interviews, practicioners often use node-link diagrams for visual sanity checking to gain a quick overview of a KG. Here, scalability issues arise,

though: first, node-link diagrams can be difficult to parse if the KG is dense, i.e., if the nodes exhibit a high degree. Second, large KGs can become difficult to render computationally.

Apart from this, end-users can also not be expected to extract information of interest from node-link diagrams with thousands, millions, or billions of nodes. Some interviewees even report that their end-users actually prefer table-based presentation formats over node-link diagrams. The authors state that reasons for this could be the simplicity and familiarity of tables across multiple domains and the fact that the tasks of end-users are often straightforward in comparison to the more complex analysis tasks of experts.

In the remainder of their paper, the authors present future research directions for the visualization of KGs that can help mitigate these challenges. However, these early-stage ideas do not suffice as concrete recommendations to be applied in practice. This leaves the question why none of the works examined above provide concrete recommendations for the utilization of KGs but only discuss opportunities and in parts challenges. Especially considering the present hype around KGs, recommendations that provide guidance for both new and experienced practicioners is pivotal for the maturation of the technology. As it turns out, the answer is that surveys and other meta-level works that focus on a bigger picture are simply not an appropriate way of communicating recommendations. The reason for this is that recommendations mostly apply to specific KG topics rather than KGs as a whole. Hence, there are works that, for example, cover best practices specifically for the topic of KG creation such as the chapter by Rashid et al. in a report of a recent Dagstuhl seminar (Bonatti et al., 2018). To handle this problem in this thesis, the following section introduces a categorization of KGs based on their scope such that subsequent sections can express the applicability of opportunities, challenges, and recommendations identified herein more precisely.

2.3 Scope and Size of Knowledge Graphs

Previous work on the utilization of KGs indicates that there is bidirectional relationship between KGs and their applications: On the one hand, KGs are initially created with some tasks or applications in mind that determine their scope and content. On the other hand, established KGs also spark, with respect to the possibilities their scope permits, ideas for new applications such that they are leveraged for other purposes. To distinguish the applications discussed in this thesis based on the scope of the employed KGs, the KGs are divided into three categories in terms of their scope's broadness:

- Open-domain KGs include entities from any domain. To be able to encode the relationships between these heterogeneous entities, they employ a large set of properties with a varying degree of specificity from an arbitrary number of vocabularies. Examples for this category are Wikidata (Vrandecic & Krötzsch, 2014), DBpedia (Auer et al., 2007), Freebase (Bast et al., 2014), and YAGO (Tanon et al., 2020). Wikidata, for instance, uses over 10,000 proprietary properties from proprietary vocabularies[30] such as the generic property P31 (instance of) or the rather specific property P9652 (personality trait of fictional character) in addition to common RDF vocabularies including SKOS (Miles & Bechhofer, 2009). Application 1, which is discussed in Section 3, utilizes KGs from this category.

- Domain-specific KGs are more confined as they only permit entities of a certain domain or subdomain. Accordingly, the employed vocabularies focus on properties that are useful for the respective domain. KGs such as KnowLife (Ernst et al., 2014), a KG for the health and life sciences, and the Open Research Knowledge Graph (ORKG) (Jaradeh et al., 2019), a knowledge graph integrating the scientific information communicated in research papers, fall into this category. Accordingly, KnowLife comprises, for example, a property expressing a contraindication (Ernst et al., 2014) while the ORKG uses properties such as P59028 (has research problem)[31]. Application 2, which is discussed in Section 4, utilizes KGs from this category.
- KG fragments are minimal KGs that encompass concise pieces of knowledge. Accordingly, they only use a small number of properties. An example for this category are nanopublications (Kuhn et al., 2018), a format for scientific publications that encodes contributions in RDF format to facilitate a direct integration with Scientific Knowledge Graphs (SKGs). The Applications 2 and 3, which are discussed in Section 4 and Section 5, utilize KGs from this category.

Previous work also shows that the scope broadness influences the diversity of conceivable applications: While Wikidata has been employed for applications as diverse as the computation of inter-document similarity, image classification, and the curation of genomic data (Cantallops et al., 2019), among many others, applications leveraging the ORKG are mostly limited to research on scholarly communication.

Apart from the scope broadness, the size of a KG also affects potential applications to a certain degree. Clearly, small KGs might not be useful for data intensive ML applications. But, there is also the other extreme: The Wikidata Query Service, for example, struggles to respond to various, in many cases not even complex, SPARQL queries due to the vast amount of data within Wikidata[32]. Hence, the size of a KG can impede its applicability under certain circumstances, as well. In this context, it is also important to note that the scope of a KG does not automatically allow conclusions to be drawn about its order or size. However, one can usually observe a positive correlation between the broadness of the scope and the number of present entities and relations, at least for well-maintained KGs. The reason for this is straightforward: If the scope does not exclude groups of entities, there are more entities that can be included, increasing the KG's order. Additionally, if there are more properties permitted, more entity relationships can be encoded, which ultimately increases the KG's size. The following sections, i.e., Section 3, Section 4, and Section 5, which discuss the three principal applications examined in this thesis, will provide more insights regarding the relationship between KGs and their applications.

[30] https://www.wikidata.org/wiki/Wikidata:Database_reports/List_of_properties/all (visited 2024-07-11)
[31] https://orkg.org/property/P59028 (visited 2024-07-11)
[32] Section 3 elaborates on this issue.

3 Generation of Dual-Entity Knowledge Panels (Application 1)

In order to answer the users' information need faster, the result pages of modern web search engines such as Startpage[33], Ecosia[34], Google[35], and Bing[36] present additional box-shaped interface elements, called Knowledge Panels (KPs), with supplementary information on entities mentioned in the search queries. These interface elements are typically located on the right of the traditional vertical list of search results. Originally, they have been introduced by Google as one of the main technologies powered by their new KG (Singhal, 2012), before the competitors released their own versions. Recently, Bing reimagined their variant, calling them Knowledge Cards 2.0 now (Bing, 2023). The entity information displayed in KPs is retrieved from a KG. For this purpose, the search queries are processed by means of entity linking techniques to identify mentions of entities in queries and link them to entities in the employed KG. Afterwards, the KP is constructed based on the information that the KG provides for the identified entities. Table 4 gives an idea of the diverse entity information presented in KPs.

What motivates the further investigation of this topic, is that KPs are only reliably displayed by the examined search engines if the respective query mentions exactly one entity. Such queries are herein called single-entity queries. When dual-entity queries, which are queries that mention exactly two entities, are issued, the quality of the KPs decreases: In some cases, no KP is displayed at all, in other cases, a KP is created for only one of the entities and sometimes the presented KP only partially matches the

Table 4: A subset of the information displayed in the knowledge panel variants of Google, Bing, and Startpage when the two queries *European Union* and *Alan Turing* are issued individually. Information from third-party sources like weather services has been left out. The search engines were set to English and the searches were conducted on 2023-05-26. Table and caption adapted from [II].

Query: *European Union*			Query: *Alan Turing*		
Knowledge Panel of			Knowledge Panel of		
Google	Bing	Startpage	Google	Bing	Startpage
Area	Description	Description	Occupation	Occupation	Description
Founding Date	Capital	Motto	Born	Born	Born
Founders	Largest metropolis	Anthem	Died	Died	Died
Awards	Official languages	Capital	Movies	Cause of death	Cause of death
Subsidiary	Official scripts	Institutional seats	Influenced by	Education	Education
	Religion	Largest metropolis	Siblings	Alma mater	Alma mater
	Demonym(s)	Official languages	Awards	Known for	Known for
	...	...		...	...

[33] https://www.startpage.com (visited 2024-07-11)
[34] https://www.ecosia.org (visited 2024-07-11)
[35] https://www.google.com (visited 2024-07-11)
[36] https://www.bing.com (visited 2024-07-11)

mentioned entities[37]. In this regard, an argument can be made that this is a missed chance since KPs for dual-entity queries could explain the relationship between the two mentioned entities in order to potentially satisfy the users' information need without requiring them to consult the ranked search results. Hence, the first application discussed in this thesis encompasses the implementation of dual-entity KPs that provide a meaningful explanation of the semantic relationship between the pair of entities mentioned in dual-entity queries. To do this, a path that connects the entity pair in the employed KG is retrieved and used as the foundation for generating what is called an entity relationship explanation in the literature (Reinanda et al., 2020). Clearly, the meaningfulness of an entity relationship explanation is highly subjective as it depends on the user's information need and preknowledge, the search context, the path length, included properties and entities, and many other factors. In fact, there is an entire discipline called information seeking concerned with investigating the subjective value of information in the context of (web) search (Choo et al., 2000). The thorough assessment of all these factors exceeds the scope of this thesis as it necessitates large-scale user studies in the future. Herein, the assumption is made that the meaningfulness of paths can be interpreted as the absence of concept drift (Dietz et al., 2018). Concept drift arises when the semantic scope of the query is left or, in other words, when the entities on a path exhibit a high semantic distance to each other. Consider the following two paths, connecting the entities *Moon* (Q405) and *Milky Way* (Q321) in Wikidata:

Moon (Q405) —described by source (P1343)→ Brockhaus and Efron Encyclopedic Dictionary (Q602358)
←described by source (P1343)— Milky Way (Q321)

Moon (Q405) —parent astronomical body (P397)→ Earth (Q2) —part of (P361)→ Solar System (Q544)
—part of (P361)→ Milky Way (Q321)

In this case, the first path exhibits concept drift by traversing an entity representing an encyclopedia whereas the second path includes entities that are semantically more related to the two entities from an intuitive point of view. The second path is thus considered more meaningful, here. How this interpretation of meaningfulness affects the quality of the found paths, how the semantic distance between entities can be computed, and how paths can be transformed into entity relationship explanations is explained in the following sections.

Since users are expected to issue queries mentioning arbitrary entities from virtually any domain to the big web search engines from above, only large open-domain KGs qualify for the implementation of dual-entity KPs. Otherwise, the entities entered would often not be found in the KGs, which would prevent the pathfinding and thereby also the creation of entity relationship explanations. At the same time, search engines are a time-critical application domain where the swift display of search results is a pivotal requirement. Combined with the demanding KG scope and size, this poses significant challenges for the implementation, resulting in an interesting application regarding the goals of this thesis. The following sections show that employing search heuristics with focus on the semantic distance between entities not only helps to avoid concept drift but also accelerates the pathfinding.

[37] Examples for this observation are presented in [II].

3.1 Content and Positioning of Paper I

The next content is contained in [I] (Martin et al., 2020). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information and details on my contributions to this paper can be found in the back of this thesis.

3.1.1 Key Contributions

Paper [I], the initial paper addressing the implementation of dual-entity KPs, introduces the first version of the devised pathfinding algorithm. To be able to quickly find paths between a source entity e and a target entity e' in Wikidata, i.e., the chosen open-domain KG, the bidirectional A^* search algorithm shown in Listing 4 is employed. Like any pathfinding algorithm based on A^* search (Hart et al., 1968), the candidate nodes, in our case entities, to be visited next are ranked in a priority queue. The ranking is informed by some search heuristics that is tailored to the specific search problem.

In the context of Application 1, the search heuristics takes the semantic distance between entities into account. In [I], the semantic distance between two entities is calculated based on the cosine distance[38]

procedure FindPath $(e, e', \alpha, \beta, \gamma, entityLimit)$:
1 $priorityQueue \longleftarrow \langle e, e' \rangle$
2 $reachable_{source} \longleftarrow \{e\}$
3 $reachable_{target} \longleftarrow \{e'\}$
4 $visitedEntities \longleftarrow \{\}$
5 **while** $priorityQueue \neq \emptyset$ **and** $|visitedEntities| < entityLimit$:
6 $entity \longleftarrow dequeue(priorityQueue)$
7 $visitedEntities \longleftarrow visitedEntities \cup \{entity\}$
8 **if** $entity \in (reachable_{source} \cap reachable_{target})$ **then**
9 **return** $reconstructPath(e, e'), |visitedEntities|$
10 **end if**
11 **for** $adjacentEntity \in getAdjacentEntities(entity)$ **do**
12 $costs \longleftarrow calculateCosts(e, e', adjacentEntity, \alpha, \beta, \gamma)$
13 $enqueue(priorityQueue, adjacentEntity, costs)$
14 **if** $entity \in reachable_{source}$ **then**
15 $reachable_{source} \longleftarrow reachable_{source} \cup adjacentEntity$
16 **else if** $entity \in reachable_{target}$ **then**
17 $reachable_{target} \longleftarrow reachable_{target} \cup adjacentEntity$
18 **end if**
19 **end for**
20 **end while**
21 **return** $\perp, |visitedEntities|$

Listing 4: The bidirectional A^* search algorithm from [I]; more verbose and with adapted notation as presented in [II].

[38] Cosine distance is a derivative of cosine similarity. In this thesis, the cosine distance is calculated using the simple formula $CosineDistance = 1 - CosineSimilarity$, which suffices here.

between the fastText (Bojanowski et al., 2017) word embeddings of their english labels[39]. The consideration of the semantic distance has two benefits: First, when resulting paths comprise entities that are semantically related and thereby avoid concept drift (Dietz et al., 2018), they are expected to be perceived more meaningful. Second, the observation was made that, on average, the semantic distance between entities increases with their hop distance. Hence, entities in close proximity typically exhibit a lower semantic distance. Visiting entities with a low semantic distance to the target entity is therefore expected to accelerate the pathfinding process. Additionally, the search heuristics also considers the path length since paths that are too long are not accessible for users and cannot be presented effectively.

Taking into account the general cost function $f(p) = g(p) + h(p)$ of the A^* search algorithm (Hart et al., 1968), the *calculateCosts* function depicted in line 12 of the algorithm calculates the costs associated with a path p comprising n entities using the following equations from [I][40]:

$$g(p) := \alpha \cdot \bar{d}\left(p_{[..n-1]}, e'\right) + \beta \cdot n$$

$$h(p) := \gamma \cdot d(v_n, e')$$

where $p = \langle e, ..., v_n \rangle$

and $p_{[..f]}$ is the sub-path $\langle e, ..., v_f \rangle$ of p.

The hyperparameters α, β, and γ used in these equations and Listing 4 serve as the weights of the cost function components. In $g(p)$, the function $\bar{d}\left(p_{[..n-1]}, e'\right)$ calculates the average semantic distance between each entity on the path excluding the last one and the target entity, which is then added to the path length n. In $h(p)$, the function $d(v_n, e')$ calculates the semantic distance between the last entity of a path and the target entity.

Paper [I] presents a preliminary evaluation of the algorithm based on four intuitively set hyperparameter configurations and a small set of hand-selected pairs of entities from Wikidata. The results show that hyperparameter configurations that use the cost function components considering semantic distances can beat a baseline configuration, which mimics breadth-first search by having set α and γ to 0 and β to 1, in terms of the number of entities that have to be visited to find a path.

3.1.2 Positioning of the Paper

Regarding opportunities that arise from the utilization of KGs, the application presented here employs the diverse and rich information within large open-domain KGs to increase the utility of search engines, which represent a well-established and heavily researched application domain. The information employed does not only comprise symbolic factual knowledge about entities directly attached to them but also the intricate relationships between entities. In particular, relationships between data points cannot be retrieved as easily using other data management solutions such as relational databases. The fact that Google's competitors quickly introduced their versions of KPs to their search engines demon-

[39] This approach has downsides which are addressed in [II].

[40] The equations shown here are slightly modified to comply with the notation used in Listing 4.

strates the impact of the technology. An extension from presenting the key facts about single entities to explaining the relationship between two entities appears to be a natural extension. This also aligns with ideas underlying the semantic web where the semantics of raw information in the web are extracted and employed for other end-user applications (Hogan, 2020).

From an algorithmic standpoint, the implementation of dual-entity KPs poses significant challenges due to the scope and size of Wikidata and the requirements of modern web search engines in terms of runtime performance. Paper I demonstrates how to mitigate these challenges through effective restrictions and the exploitation of available information. In this regard, one salient aspect is the bidirectionality introduced to the pathfinding algorithm, which is not part of standard A^* search: To retrieve the entity information including entity labels and adjacent entities from Wikidata, the pathfinder leverages the SPARQL-based Wikidata Query Service[41]. Originally, the idea was to retrieve adjacent entities by following both outgoing and incoming edges of an entity, effectively treating the KG as if it was a bidirectional graph. This is possible because properties can be interpreted in both directions by humans, as explained in Section 2. For example, a property labeled *instance of* pointing from an entity e_i to an entity e_j can be interpreted as a semantically equivalent inverse property labeled *has instance* that points from e_j to e_i. A valid path from e to e' can therefore comprise properties with alternating directionality in this application. Hence, the following path connecting the entities *Bamberg* (Q3936) and *Bamberg University Library* (Q23786596) also qualifies as a valid path and could thus serve as a basis for generating an entity relationship explanation to be presented in a dual-entity KP:

Bamberg (Q3936) —instance of (P31)→ college town (Q1187811) —has facility (P912)→ university (Q3918) —subclass of (P279)→ public university (Q875538) ←instance of (P31)— University of Bamberg (Q707272) —has part(s) (P527)→ Bamberg University Library (Q23786596)

However, due to Wikidata's ontology, there are entities exhibiting a vast number of incoming edges, while the number of outgoing edges is far lower most of the times. For instance, the entity *Germany* (Q138) has 3,366,737 incoming edges in contrast to 2,776 outgoing edges[42]. The number of incoming edges is partly so high that the Wikidata Query Service is not able to return all adjacent entities before timing out. The strain vast amounts of data can put on interfaces is a challenge that translates to all large-scale KGs and is also known in big data in general. In this instance, the interface is maintained by a provider such that it cannot be altered. Hence, one has to find workarounds to tackle this problem for the present application. To this end, the decision was made to only pursue outgoing edges during pathfinding but to conduct two searches simultaneously, one from e to e' and one from e' to e. The underlying assumption that Wikidata is meshed enough such that this restriction does not impede the success rate of the pathfinding is confirmed in the study conducted in II.

[41] https://query.wikidata.org (visited 2024-11-26)

[42] Data retrieved using the queries available at https://w.wiki/AgrY and https://w.wiki/Agre on 2024-07-11. As said, the large discrepancy between incoming and outgoing edges is caused by Wikidata's ontology. For example, non-person entities that exist in a country comprise the property *country* (P17) pointing to the respective country entity. If all instances of this property would be replaced by an inverse property *contains*, the proportions of incoming and outgoing edges of country entities would be significantly altered due to an increase of outgoing edges.

While the preliminary evaluation in [I] indicates that limiting the edges to be pursued solves the SPARQL interface problem, it also shows that this method alone does not sufficiently address the runtime performance requirement of web search engines, as the hyperparameter configuration mimicking breadth-first search still has to visit too many entities to find paths. The other examined hyperparameter configurations successfully exploit entity labels as an additional information source to accelerate the pathfinding. This demonstrates how information that is not directly linked to a specific problem can be exploited under certain circumstances. Regarding the utilization of large open-domain KGs, which accommodate lots of heterogeneous data by design, this is an important insight.

Clearly, it is still important to ensure that the exploited information fits the problem. For instance, one could intuitively think that KG embeddings (Wang et al., 2017) should be used as search heuristics here since they are specifically trained on the target KG and can thus facilitate computing shortest paths between entities within it. In the present application, the goal of the pathfinding is, however, not actually to find the shortest path in terms of the hop distance. Instead, the goal is to satisfy the users' information need by presenting meaningful entity relationship explanations. Again, consider the two paths, connecting the entities *Moon* (Q405) and *Milky Way* (Q321) from above. Even though the second path is longer than the first one, the entities on the path are semantically more related to each other than the entities on the first path from an intuitive point of view. Hence, an argument can be made in favor of the second path considering the goal of providing meaningful entity relationship explanations. Search heuristics solely focusing on the path length are therefore less effective here. That being said, combined with other components such as the semantic distance from above KG embeddings could also be considered for the cost function in the future. However, this again implies new and significant challenges such as the computation of KG embeddings on KGs as large as Wikidata. The approach from above uses of-the-shelf fastText word embeddings, thereby eliminating the need for costly training.

3.2 Content and Positioning of Paper II

The next content is contained in [II] (Martin, 2023). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information and details on my contributions to this paper can be found in the back of this thesis.

3.2.1 Key Contributions

Despite the promising results of the preliminary evaluation presented in [I], the paper still left several open questions to be tackled in [II]. The first point to be addressed is related to the computation of the semantic distance between entities. In [I], the semantic distance is calculated solely based on the cosine distance between word embeddings of the english entity labels. However, these labels are highly ambiguous, as there are many entities in Wikidata that possess similar or even identical labels. To mitigate this problem, the computation of the semantic distance is adapted to also take entity descriptions into account, another source of information available in Wikidata. These entity descriptions are

provided through the `https://schema.org/description` properties attached to the entities. To compute the embeddings, entity labels and descriptions are concatenated and fed into SBERT (Reimers & Gurevych, 2019), a state-of-the-art transformer-based approach for calculating vector representations of sentences[43]. The first line in Table 5 shows that the additional consideration of entity descriptions confirm the semantic distances where the entity labels alone are expressive enough, while the pairs of lines two and three as well as four and five show how this approach helps to disambiguate entities with similar or identical labels, yielding more accurate semantic distances.

In I, the pathfinding implementation is evaluated based on a small hand-selected set of dual-entity queries. This set is neither representative nor suitable for conducting an optimization of the cost function hyperparameters α, β, and γ, not to mention a proper evaluation. That being said, previous work does not propose datasets for benchmarking pathfinding in Wikidata, making it necessary to create such a dataset first-handedly. To this end, II introduces a process for deriving dual-entity queries from the Million Query Track of TREC 2007[44], which comprises 10,000 realistic textual queries for web search engines. In this process, the state-of-the-art entity linker GENRE (Cao et al., 2021) is employed

Table 5: Comparison of semantic distances using five examples with two Wikidata entities each. d_{labels} and $d_{labels+descs}$ denote whether the presented semantic distances were calculated using the cosine distance between SBERT vector representations of the entity labels alone or of the concatenated entity labels and entity descriptions. Table and caption adapted from II.

Entities	Entity Labels	Entity Descriptions	d_{labels}	$d_{labels+descs}$
Q30	United States of America	country in North America	0.720	**0.793**
Q47488	International Criminal Court	intergovernmental organization and international tribunal		
Q243	Eiffel Tower	tower located on the Champ de Mars in Paris, France	**0.511**	0.499
Q90	Paris	capital and most populous city of France		
Q243	Eiffel Tower	tower located on the Champ de Mars in Paris, France	0.511	**0.758**
Q167646	Paris	mythological son of Priam, king of Troy		
Q6004986	Immigration	album by Show-Ya	0.402	**0.786**
Q841440	naturalization	process by which a non-citizen in a country may acquire citizenship or nationality of that country		
Q131288	immigration	movement of people into another country or region to which they are not native	0.402	**0.451**
Q841440	naturalization	process by which a non-citizen in a country may acquire citizenship or nationality of that country		

[43] Apart from the generally superior performance of SBERT, the replacement of fastText with SBERT is necessary since fastText is designed for embedding individual tokens while entity descriptions consist of multiple words.
[44] https://trec.nist.gov/data/entity.html (visited 2024-11-26)

to identify mentions of entities in the TREC queries and link them to Wikidata entities. The resulting dataset consists of 1,196 unique entity pairs.

The established dual-entity query dataset made it possible to conduct a hyperparameter optimization for fitting the hyperparameters α, β, and γ. The goal of the optimization is to find a hyperparameter configuration that minimizes the average number of entities that have to be visited during pathfinding to find a path (and *not* to find a configuration that finds paths with minimal length). For this purpose, the Simple(x) optimizer[45] is employed to test different configurations based on 20% of the query dataset, i.e., 239 dual-entity queries. Figure 7 shows the results of the optimization. In the diagram, the *objectiveValue* is the average number of visited entities achieved by the hyperparameter configuration examined in the respective iteration. The *minValue* equals the minimal *objectiveValue* encountered so far. As depicted, a good *minValue* is already achieved after 20 iterations. Overall, the best result is encountered in the 101st iteration using the hyperparameter configuration $\alpha = 0.699$, $\beta = 0.109$, and $\gamma = 0.823$. A significant improvement beyond the 150 considered iterations is not expected.

Apart from the hyperparameter optimization, the dual-entity query dataset also allows for a proper evaluation of the approach. For this purpose, the pathfinding implementation, which has been fully re-implemented for [II] and is now called BiPaSs, is tested on the unseen 80% of the query dataset, i.e., 957 dual-entity queries. Regarding hyperparameter configurations, the four original options from [I], which include the baseline breadth-first like configuration and the optimized configuration from above, are considered. The optimized configuration achieves a test set coverage of 79.2%, while the baseline configuration is able to find a path in only 55.6% of cases. Solely the runner-up configuration, called Semantics-Only in [I] and [II], which only takes the cost function components into account that consider the semantic distances, achieves a comparable coverage of 73.6%. However, since this config-

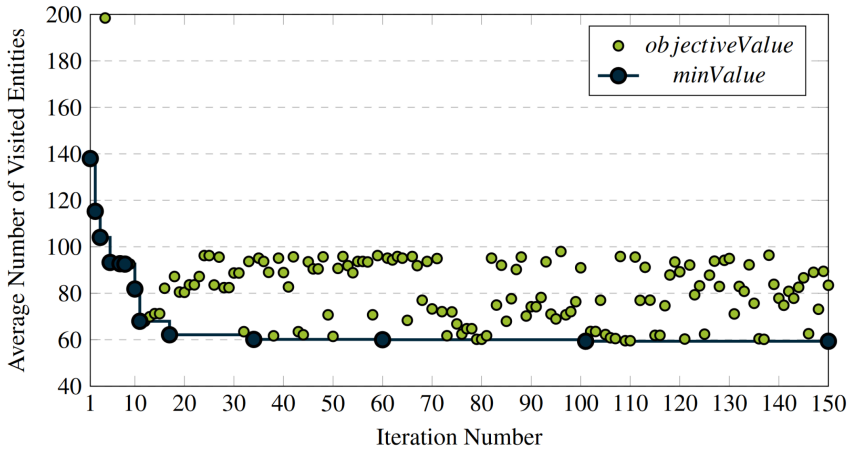


Figure 7: The results of the hyperparameter optimization. The y-axis denotes the average number of entities visited by the configuration examined in the respective iteration, i.e., the *objectiveValue*. The *minValue* equals the minimal *objectiveValue* encountered up to the respective iteration. Figure adopted from [II].

[45] https://github.com/chrisstroemel/Simple (visited 2024-11-26)

uration neglects the path length, the resulting paths have a length of about ten on average, which raises concerns regarding their utility for users. Furthermore, longer paths are more difficult to present given the space constraints of KPs. In comparison, the paths found by the optimized configuration exhibit a moderate length of about five on average.

3.2.2 Positioning of the Paper

As a direct follow-up to [I], [II] elaborates on the idea of exploiting the rich information from KGs for other applications. At the same time, it also highlights the importance of a careful examination of the leveraged information. In fact, the original naive approach of taking only entity labels into account, which are in many cases ambiguous, during the computation of semantic distances can significantly affect the performance of the pathfinding: When entities that exhibit an alleged low semantic distance are prioritized even though they are not actually semantically related to the other entities, the path quality plummets due to concept drift. Combined with a thorough understanding of the application, the careful inspection of the available data and the problems that come with it is thus crucial for its effective exploitation. This practice is foundational for effective data science (Provost & Fawcett, 2013), showing that principles from this discipline apply to the KG domain.

The vast amounts of heterogeneous data within large KGs and their non-uniform structure make it difficult to obtain an overview and to identify trends and tendencies. However, the application investigated here requires efficient pathfinding which depends on the usage of expedient search heuristics. The idea that the semantic distance between entities tends to correlate positively with their hop distance from [I] is an assumption that was made based on a relatively small number of examples. Only in the subsequent paper [II], the hyperparameter optimization and the evaluation of the algorithm using the novel dual-entity query dataset confirm the utility of search heuristics that build upon this assumption. In the worst case, these activities could have also revealed that the assumption does not apply to entities in Wikidata in general, requiring a complete reassessment of the search heuristics. This incident demonstrates that a creative and exploratory approach can be an effective strategy to tackle problems associated with large KGs, at least if the risk is acceptable in the given context.

Papers [I] and [II] mainly discuss the number of visited entities and the coverage of the test set as objective metrics to assess the performance of the pathfinding algorithm, the employed search heuristics, and the hyperparameter configurations. At the same time, found paths exhibit characteristics that affect the perceived path quality and therefore also their utility regarding the envisioned dual-entity KPs. These characteristics comprise the path length, the comprehensibility of the relationships between the entities, and the understandability of properties, among others. How these factors affect the perceived utility of paths has not been addressed yet. Especially the assumption that the consideration of the semantic distances not only facilitates finding paths quickly but also finding paths that users perceive as meaningful has to be tested. For this reason, a large-scale user study has to be conducted in the future.

3.3 Content and Positioning of Paper III

The next content is contained in [III] (Martin & Henrich, 2023a). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information and details on my contributions to this paper can be found in the back of this thesis.

3.3.1 Key Contributions

The previous sections addressed the identification of meaningful paths that could serve as a basis for generating entity relationship explanations. Paving the way towards an end-to-end implementation of dual-entity KPs, [III] approaches the question of how the found paths can be presented in KPs as entity relationship explanations. But, the main objective of this paper is not yet the discussion or evaluation of production-ready presentation formats. Instead, the introduction of technology facilitating the exploration of these topics lies in focus here. The said technology encompasses a testbed for the implementation and evaluation of conceivable presentation formats. Note that the term presentation formats includes graphical visualizations but also approaches that, for example, verbalize the found paths.

Figure 8 provides a screenshot of the testbed. As depicted, users are able to select the two entities of a dual-entity queries and a presentation format to be applied. Since there is no connection to the BiPaSs

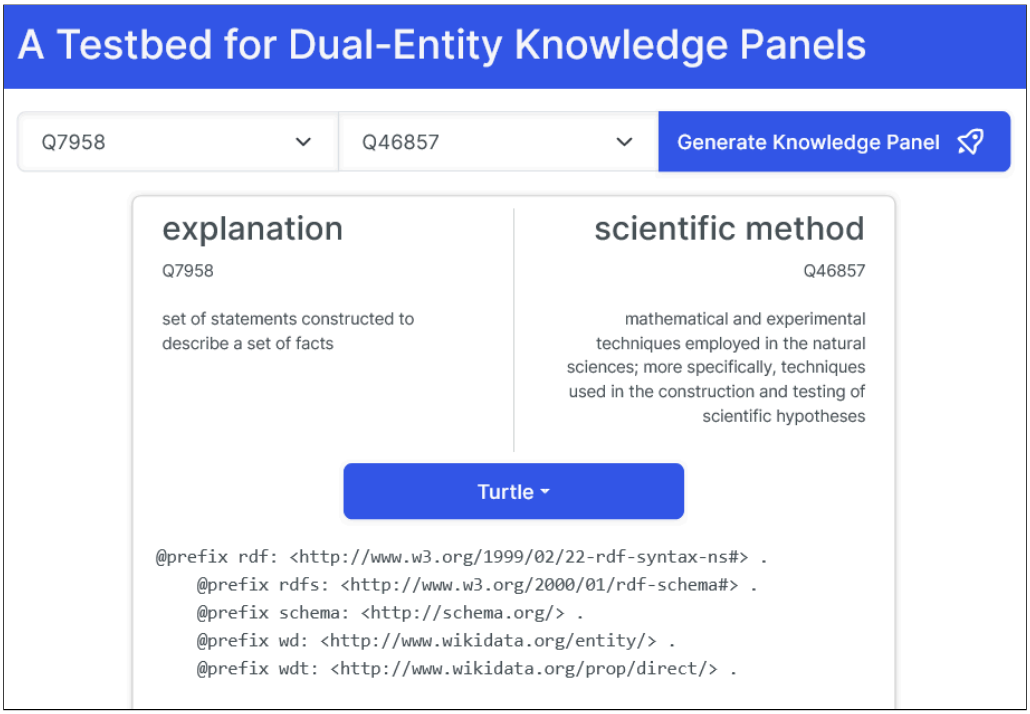
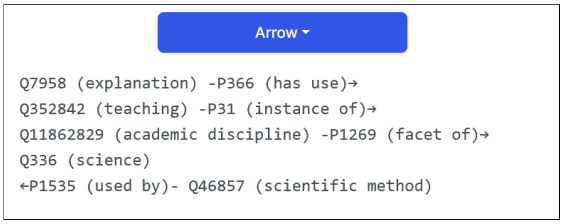


Figure 8: The user interface of the testbed after clicking the *Generate Knowledge Panel* button. In this screenshot, the lengthy Turtle document encoding the path is not shown completely. Figure and caption adopted from [III].

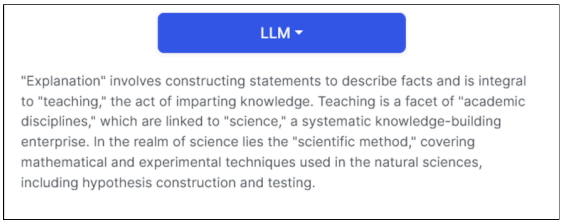
pathfinding implementation yet, only a limited set of entity combinations is available in the testbed. By default, the *Turtle* presentation format is selected, which is a placeholder for debugging, as it displays the raw path encoded in Turtle. To promote further research, [III] additionally introduces the four prototypical presentation formats shown in Figure 9. The *Arrow* presentation format is similar to the path notation used within this thesis. Demonstrating the potential of combining KGs and LLMs, the *LLM* presentation format shows a verbalization of the Turtle-encoded path generated using ChatGPT 3.5 and appropriate prompts. The remaining two presentation formats *Graph: Circle* and *Graph: Hierarchy* both use a graph-based visualization of the path, i.e., node-link diagrams, but with different layouts. Hovering over the labels of entities and properties reveals their Wikidata IDs and descriptions. Intuitively, one could make an argument in favor of the hierarchical approach of the *Graph: Hierarchy* presentation format, as it takes the directionality of the properties into account. The further development of the presentation formats shown above but also of other presentation formats, must be based on user studies to be able to test such assumptions.

3.3.2 Positioning of the Paper

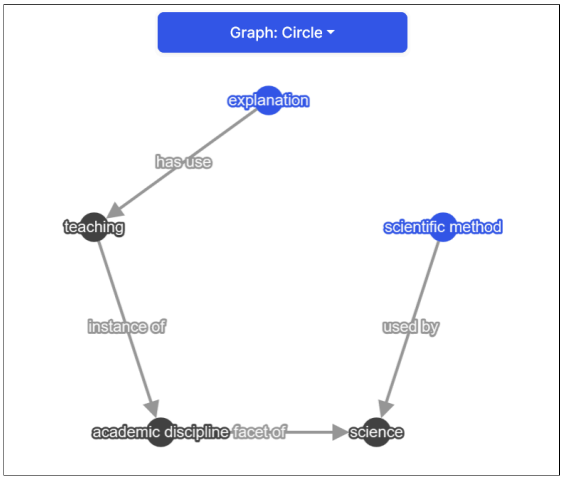
Due to its generic data model, RDF data can be visualized and presented in a variety of ways. In fact, the presentation formats mentioned above only reflect a small subset of conceivable options. Graph



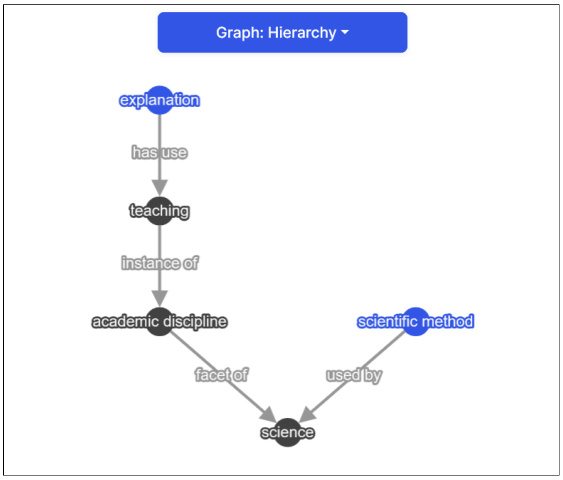
(a) The *Arrow* presentation format.



(b) The *LLM* presentation format.



(c) The *Graph: Circle* presentation format.



(d) The *Graph: Hierarchy* presentation format.

Figure 9: The four currently implemented prototypical presentation formats. Figures and captions adopted from [III].

visualization, in particular, is a discipline with a long tradition (Herman et al., 2000). However, the present application calls for path visualization or, more generally, presentation techniques, which is a less explored topic. Considering the search engine context, the development of easy to understand yet expressive presentation formats is pivotal for the implementation of useful dual-entity KPs conveying entity relationship explanations, though. Large open-domain KGs offer a lot of heterogeneous information that could be used to enrich presentation formats. This includes entity and property descriptions, taxonomic relationships, multi-lingual labels, adjacent entities that are not part of the path, and many more, thus necessitating iterative user studies for the further development of presentation formats[46].

From a software engineering perspective, the testbed demonstrates an interesting advantage of RDF's inherent serialization capabilities. Leveraging the Turtle format, the path information is supplied to the testbed directly in code without dependencies to the BiPaSs pathfinding implementation. Later on, an interface can be added to the testbed to establish a connection to a back end running BiPaSs that readily supplies the serialized paths to the front end. Following the client server architecture, this interface can also be implemented as a web API since the RDF serialization formats can be easily exchanged via HTTP. Ultimately, RDF's serialization capabilities streamline the development of individual software components and interfaces, facilitating the construction of complex distributed systems.

3.4 Presentation Formats for Entity Relationship Explanations

As a continuation of the presented research, the bachelor theses by Simon Brand (Brand, 2023) and Jeremias Udaly (Udaly, 2023), which were supervised by me, elaborate on the investigation of suitable presentation formats. In this context, the thesis by Simon Brand focuses on different options for the graphical visualization of entity relationships. The first key insight presented in his work relates to the thematic focus of the paths to be visualized. While the context necessitates a generic fallback presentation format applicable to all paths, other topic-specific presentation formats are conceivable that excel in the presentation of paths within a certain thematic focus. For example, map-based visualizations could be a useful option when all entities of a path are geographic entities. In contrast, when all entities are animals and the properties signify some biological relationship one could depict the entity relationship as a phylogenetic tree. However, BiPaSs currently does not explicitly look for paths that adhere to a certain thematic focus. Due to the high diversity of the entities in Wikidata, it is therefore mandatory to implement a generic presentation format that can be applied to paths with entities and properties of an arbitrary thematic focus. Hence, the central contribution of the bachelor thesis encompasses the development and evaluation of different generic visualizations using the testbed from III. Interestingly, the test persons rated node-link diagrams, which are a standard option for visualizing graph structures, the highest. At the example of

[46] In her master thesis (Thaller, 2021), which was supervised by me, Theresa Thaller tackled the problem of displaying and navigating large RDF-based KGs and came to a similar conclusion. Readers interested in the visualization of RDF graphs in contrast to the presentation of individual paths from RDF graphs, which is investigated here, are therefore kindly referred to her thesis.

explanation (Q7958) —has use (P366)→ teaching (Q352842) —instance of (P31)→ academic discipline (Q11862829) —facet of (P1269)→ science (Q336) ←used by (P1535)— scientific method (Q46857)

as a path, Figure 10 shows the final node-link diagram prototype. As depicted, this presentation format displays the entities as circles which are connected via bars whose width and color corresponds to the semantic distance. A green thick bar indicated that the associated entities are more closely related than a thin red bar. Additionally, entities with a higher semantic distance are positioned further apart than entities with a lower semantic distance. The property labels are attached to the bars via dotted lines. There is also an additional button that reveals additional information on the meaning of the used colors when clicked. This presentation format demonstrates the variety of aspects that have to be considered regarding the implementation of dual-entity knowledge panels to maximize effectiveness and usability. Hence, there are several questions that have to be answered through additional user studies, including:

- Should the presentation format provide additional information from the KG related to the path such as entity descriptions or other adjacent entities?
- How should the semantic distance between entities be visualized?
- How should the entities be connected to reflect the directionality of properties[47]?

This shows that the task of visualizing single paths from a graph, which might appear relatively simple at first glance, actually poses significant challenges, especially in the context of open-domain KGs. In fact, the vast amounts of available data can be viewed as an impeding factor for this task since it leaves developers spoiled for choice. When it comes to the visualization of KG data in end-user applications, it is thus important to not be deceived by the seemingly trivial triple-based data model of RDF. The same holds for the implementation of topic-specific presentation formats, which requires similar

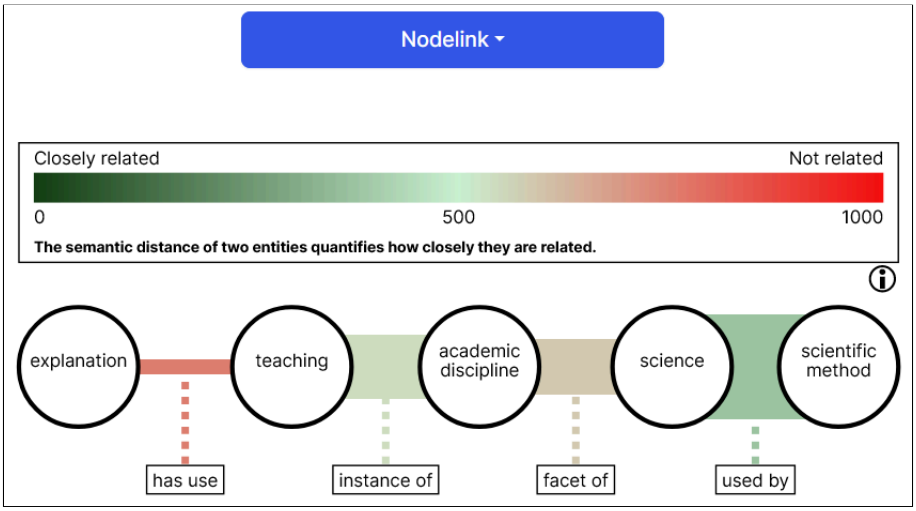


Figure 10: An example path visualized using the presentation format based on node-link diagrams as proposed in Brand (2023). The information button was clicked, revealing additional information on the meaning of the displayed colors.

[47] Note how the presentation format shown in Figure 10 omits the directionality of properties entirely.

development effort. Since paths without a thematic context are expected to occur more often, the implementation of the generic presentation format should be prioritized, though (Brand, 2023).

Apart from the presentation formats that leverage visual aids to explain entity relationships, textual presentation formats are conceivable, as well. Following this idea, the bachelor thesis by Jeremias Udaly explores the verbalization of paths using state-of-the-art LLMs, in particular ChatGPT-3.5-turbo[48] and Google Gemini 2024.02.08[49] (Udaly, 2023). Diving into the young field of prompt engineering, his thesis evaluates the verbalization capabilities of LLMs using prompts with different characteristics, considering the specific requirements of the dual-entity KP context. In particular, single-turn interaction with the LLMs of choice is compared to multi-turn interaction, the effects of different phrasing are investigated, the ability of LLMs to adhere to specified text lengths are assessed, and the occurrence of hallucinations is examined. Given the current interest in combining symbolic and subsymbolic machine learning approaches to avoid hallucinations of LLMs[50], the last point is especially interesting. As it turns out, a major factor affecting the quality of the generated verbalizations is the format, in which the paths are provided to the LLMs. Providing the paths in the Turtle format, for example, leads to the problem that LLMs sometimes miss triple statements since they are not guaranteed to be ordered in accordance with the order of the entities on the path. In contrast, providing the paths in the arrow syntax that is also used in this thesis comes with the problem that LLMs sometimes fail to understand the directionality of the properties because they have not been trained on data in this non-standardized format. That being said, the issues associated with both formats can be mitigated to a certain degree by means of prompt engineering. For instance, first tests with extended prompts indicate that providing examples on how the arrows in the arrow syntax are to be interpreted can already improve the quality of the resulting verbalizations.

Another issue discussed by Udaly (2023) is related to the LLMs' ability to follow commands specifying the content and structure of the produced verbalizations. On search engine result pages, there is, for example, only a limited amount of space for KPs. For the best user experience, it is thus important that the presentation formats adhere to certain size constraints. Regarding a textual presentation format, size constraints translate to limits of the number of characters for the verbalizations. However the results show that the considered LLMs are not able to follow instructions regarding text length appropriately. Gemini, in particular, was not able to produce verbalizations given hard character limits.

[48] https://chat.openai.com (visited 2024-07-11)
[49] https://gemini.google.com (visited 2024-07-11)
[50] In this regard, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a prominent research direction. Section 6 elaborates on this topic.

4 Import and Export of Research Contributions (Application 2)

The motivation for the second application examined in this doctoral thesis arises from the significant increase in research publications published in recent years. According to (White, 2019), the number of research publications published each year across all disciplines grew from about 1.8 million in 2008 to about 2.6 million in 2018. dblp's record complies with this observation, exhibiting a similar trend for the computer science field, as depicted in Figure 11. Hyped technologies as described in Section 1 certainly contribute to these inflated numbers. Regardless of the exact factors causing this trend, there are significant implications for the scientific community. In particular, activities such as the identification of related work, the assessment of the relevance of related work, and the extraction of relevant information from related work become increasingly time-consuming and therefore require support (Brack et al., 2020). In this regard, one key factor is that scholarly communication is still heavily document-centric today. Hence, SKGs (Luan et al., 2018), also known as scholarly knowledge graphs (Oelen et al., 2019) or research knowledge graphs (Jaradeh et al., 2019) such as the ORKG (Jaradeh et al., 2019) emerged. SKGs are domain-specific KGs that acquire and integrate scientific information in a knowledge base (Auer et al., 2018).

SKGs have the potential to fundamentally transform the way researchers acquire information for preparing their publications and the way they share their contributions. However, the envisaged shift towards what is herein called *KG augmented research* raises questions about the form of research publications, i.e., their suitability for this new research paradigm. For many decades, research publications have come in the form of self-contained documents. The shift away from document-centric research towards the envisaged KG augmented research, however, also necessitates reconsidering the form of research publications as traditional papers might not take full advantage of the new paradigm's opportunities. The supplementary papers IV, V, and VI build on this point by discussing ideas and proposing a technology to facilitate the shift away from document-centric research towards KG augmented research. Representing the second application discussed in this thesis, this technology specifically encompasses a novel publication form supporting a bidirectional knowledge exchange between

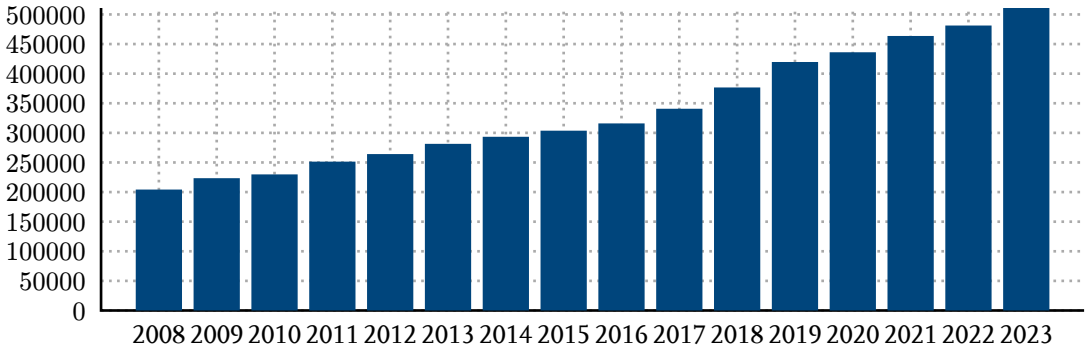


Figure 11: The number of publications added to dblp's record *per year* from 2008 to 2023. Data retrieved from https://dblp.org/statistics/publicationsperyear.html on 2024-07-03.

research publications and SKGs by means of import and export mechanisms. This application is particularly interesting as its domain, i.e., the document-centric research paradigm, is very rigid and established even though it exhibits notable flaws, which are described in the following.

4.1 Content and Positioning of Paper IV

The next content is contained in [IV] (Martin et al., 2021). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information, details on my contributions to this paper, and a preprint version can be found in the back of this thesis.

4.1.1 Key Contributions

In light of the emerging SKGs, [IV] introduces the notions of contextual and contentual information to label the different types of information SKGs can provide about research publications. Specifically, *contextual* information refers to the metadata of a research publication, whereas *contentual* information refers to its actual scientific contributions. However, only some SKGs such as the ORKG, which is RDF-based and also the most popular SKG to do so, aim to integrate contentual information of research publications, paving the ground for a wide support of research activities in the envisaged paradigm of KG augmented research[51].

The shift towards KG augmented research heavily depends on the integration of research publications into SKGs, and especially the fluency of this process. Hence, [IV] examines three available publication forms and their compatibility with SKGs from the perspectives of the research publications' authors and the maintainers of SKGs:

1. *Document-based publications*, often called papers, require authors to cover the same foundations in multiple publications even if this information is already available in SKGs. This introduces redundancy and thus inefficiency. Keywords and other classification systems only partially facilitate the integration of a paper into a SKG as they are usually not formatted in a compatible way.
2. As an attempt to narrow the gap between document-based publications and RDF, one can leverage techniques like SciIE (Luan et al., 2018), a framework for automated scientific information extraction, to generate RDF graphs from document-based publications. The original documents in combination with the generated RDF graphs are then called *RDF-transformed publications*. The problem here is that the continuously improving performance of information extraction approaches cannot mitigate the problems regarding redundancy in document-based publications since authors still have to prepare regular papers in the first place.
3. *Nanopublications* (Kuhn et al., 2018) are KG fragments that provide a native integration with RDF by directly encoding scientific contributions as RDF triples. This publication form necessi-

[51] In [IV], the paradigm was originally called knowledge graph based research. However, this notion was adjusted to KG augmented research to highlight that knowledge graphs represent a key technology but not necessarily the foundation of the paradigm.

tates training for both authors and readers, though. Hence, another solution with low impact on the current publication preparation process is required until, if at all, nanopublications become established in the scientific community.

Representing the central contribution of [IV], five requirements for a novel publication form tailored to KG augmented research are proposed:

Requirement 1 (Preparation of main contributions in natural language) The first requirement dictates that authors shall retain the ability of formulating the main contributions of their publication in natural language, like they are used to in order to minimize training effort. This requirement mainly affects authors.

Requirement 2 (Import of knowledge) In order to mitigate the redundancy problem of document-centric publications, authors shall be able import knowledge from SKGs in their publication by referencing the contributions' IRIs. This requirement mainly affects authors, as well.

Requirement 3 (Markup of knowledge) Drawing inspiration from RDFa, authors shall be able to add RDF markup to their contributions formulated in natural language. This ability simplifies the integration of contentual information into SKGs significantly. Thereby, this requirement affects both authors and SKG maintainers.

Requirement 4 (Provision of tooling for obtaining IRIs) To implement the previous requirements, in particular Requirement 2, additional tooling is required that supports authors in activities such as searching for IRIs of entities to be imported. This requirement addresses the usage context rather the envisaged publication form itself but affects authors and SKG maintainers, nevertheless.

Requirement 5 (Enriched representation of publications at view time) At view time, readers shall be presented a single self-contained document-based representation of the publication that features the knowledge imported from SKGs as specified by the authors. Hence, this requirement mainly affects readers.

4.1.2 Positioning of the Paper

In the context of the second application, the problems identified regarding available publication forms, and the requirements proposed for a publication form tailored to KG augmented research serve as the foundation for the technology proposed in [V] and [VI].

In the context of this thesis, [IV] elaborates on the opportunities that arise from the utilization of SKGs, which represent an important category of domain-specific KGs focussing on the scientific domain. At the same time, it gives a first idea of the impact on authors, SKG maintainers, and readers the introduction of an alternative publication form that provides compatibility with SKGs would cause.

Considering the rigidity of the document-centric research paradigm, the introduction of a new publication form poses significant challenges. In particular, such a publication form has to fulfil the five identified requirements to narrow the gap between papers with contributions written in natural lan-

guage, i.e., unstructured data, and RDF data[52] of SKGs like the ORKG without deterring users including authors. This is especially relevant, since connecting unstructured, semi-structured, and structured data has long been recognized as a fundamental challenge for a wide range of applications, not only in the context KGs (Blumberg & Atre, 2003). In this application, KG technology is introduced to an established system retroactively. This allows for an examination of the challenges that emerge when KGs are not part of the initial concept of an application but added later on.

4.2 Content and Positioning of Papers V and VI

The next content is contained in [V] (Martin & Henrich, 2022a) and [VI] (Martin & Henrich, 2023b). Since [VI] is an extended version of [V] that has been published in a journal's special issue, the following section summarizes the contributions of both papers relevant for this thesis jointly. Refer to [VI] for detailed information. After the summary, the papers are positioned in the context of this thesis. Bibliographic information and details on my contributions to these papers can be found in the back of this thesis.

4.2.1 Key Contributions

Typically, publishers require authors to prepare publications in compliance with guidelines that, among others, specify the permitted file formats. In computer science research, the de facto standard for typesetting publications is LaTeX[53]. Thus, the decision was made to initially focus on LaTeX-based publications regarding the implementation of a publication form tailored to knowledge graph enhanced research. Accordingly, [V] and [VI] propose RDFtex, a framework supporting a bidirectional knowledge exchange between LaTeX-based research publications and SKGs. To implement this knowledge exchange, two principal functionalities are provided:

1. The import functionality enables authors to import research contributions from SKGs in LaTeX documents by means of a custom import command.
2. The export functionality allows the export of research contributions using two custom export commands for marking up content in LaTeX documents in an RDF-compatible format.

The markup aspect of the export functionality is strongly inspired by RDFa. Accordingly, there is also another command for specifying prefixes in order to minimize the repetitive usage of namespace IRIs. Table 6 gives an overview of the, in total, four RDFtex commands. To process the custom commands within the .rdf.tex files, on which RDFtex operates, a preprocessor is employed. Files with the .rdf.tex extension are regular .tex files that moreover permit the usage of the custom RDFtex commands. The file extension was introduced to make explicit which files are to be processed via the preprocessor.

[52] Fundamentally, RDF can encode data with varying degree of structuredness, depending on how literals are used. For example, DBPedia stores entire Wikipedia sections as strings in its literal nodes but at the same time also single numeric values (cf. https://dbpedia.org/page/Bamberg; visited 2024-07-04).
[53] https://www.latex-project.org (visited 2024-07-11)

Table 6: An overview of RDFtex's four custom LaTeX commands, their purpose and the required parameters. Table and caption adapted from Martin & Henrich (2023b).

Command	Purpose	Parameters
`rdfimport`	Denotes the import of a contribution from a SciKG.	A label that will be assigned to the generated content snippet, the citation key, the IRI pointing to the contribution, and the name of the SciKG from which the contribution is to be imported
`rdfexport`	Registers an original contribution for export.	A contribution name to reference the export locally, the type of the original contribution, and an optional set of other predicates and objects that are assigned to the contribution entity if applicable
`rdfproperty`	Marks up a property of an original contribution to be exported.	The contribution name of the export, the predicate, and the content representing the object
`rdfprefix`	Registers a prefix for abbreviating a namespace or vocabulary with a prefix.	The prefix and the written out form of the namespace or vocabulary

Figure 12 illustrates the steps of the preprocessing. For a comprehensive explanation of each step in the preprocessing process refer to [VI]. In summary, the preprocessor

- replaces the import commands with content that is generated based on templates and RDF data retrieved from SKGs,
- removes the export commands while generating an export document in Turtle format that contains RDF representations of all marked up contributions, i.e., KG fragments, and
- removes the prefix commands.

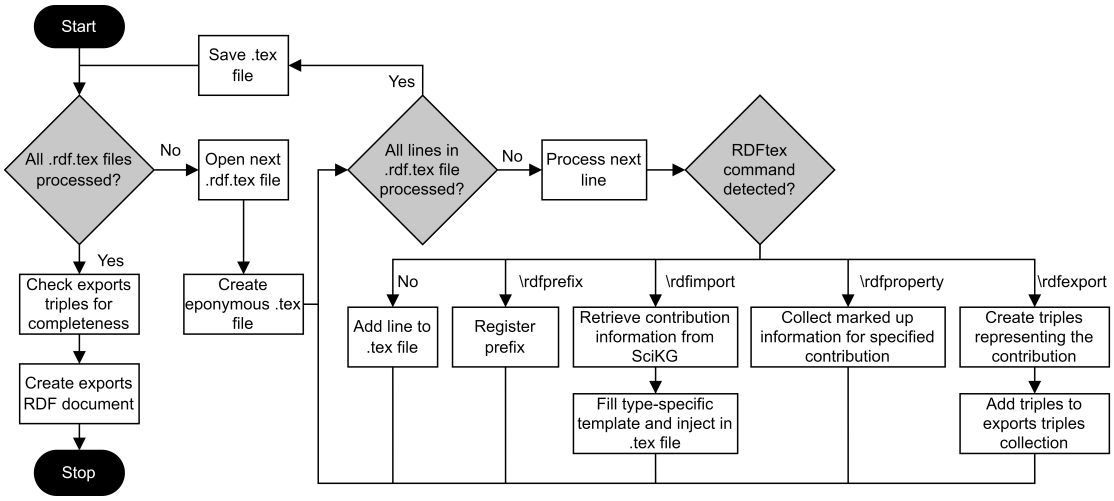


Figure 12: A flowchart of RDFtex's preprocessing workflow. For conciseness, the injection of custom LaTeX environments, which are necessary for importing some of the supported contribution types, was omitted. Figure and caption adopted from [VI].

In doing so, the preprocessor outputs standard LaTeX documents that can be compiled as usual. Currently, RDFtex supports five types of research contributions, i.e., definitions, datasets, figures, simple experimental results, and software. Each of them is associated with a set of type-specific properties. To successfully import a contribution of a given type, the target SKG has to provide all of the type-specific properties such that the preprocessor can generate LaTeX content by injecting the property values into type-specific templates. To successfully export a contribution of a given type, values for the type-specific properties have to marked up in the `.rdf.tex` files using `rdfexport` and `rdfproperty` commands. A list of the supported contribution types and the associated properties is available in RDFtex's repository[54].

Being implemented with efficiency in mind, the preprocessing only requires a single pass of all LaTeX files. In fact, experiments show that the runtime overhead introduced by RDFtex is negligible compared to the compilation duration of typical LaTeX compilers such as latexmk[55]. Furthermore, RDFtex can be used in combination with LaTeX templates that publishers prescribe. Since the export document as an artifact is completely independent from the LaTeX source files and the compiled PDF file, it can be easily integrated into SKGs, with only little involvement of SKG maintainers and publishers. These features share the common goal underlying RDFtex's implementation, namely narrowing the gap between LaTeX-based research publications and SKGs with minimal impact on established paper preparation and publication processes.

A small qualitative study was conducted with LaTeX users exhibiting varied LaTeX proficiency levels and academic degrees in computer science and astrophysics to obtain initial user feedback. While the import functionality was commended due to its simplicity and utility, the lack of appropriate tooling made the identification of relevant IRIs difficult, as expected. In this regard, VI shows how available academic search engines such as Semantic Scholar[56] can be easily adapted to provide the IRIs of research contributions. Due to the higher number of necessary RDFtex commands, the test subjects also found the export functionality to be more difficult to use than the import functionality but still managed to complete the given tasks. Several test subjects reported that they would use the technology only when the benefits of KG augmented research become tangible, though.

4.2.2 Positioning of the Papers

As a framework enabling a bidirectional knowledge exchange between LaTeX-based research publications and SKGs, RDFtex represents the second KGs application presented in this thesis. Considering the requirements set for the application selection, it is the first application to focus on domain-specific KGs, particularly SKGs. Since KG augmented research is still not widely recognized in the scientific community, the ORKG currently only provides data about 24,000 research publications[57]. As a result, it is multiple orders of magnitude smaller than the open-domain KG Wikidata used in the first application. Additionally, to showcase and evaluate RDFtex, a makeshift SKG, called MinSKG, with only a few

[54] https://github.com/uniba-mi/rdftex (visited 2024-07-04)
[55] https://ctan.org/pkg/latexmk?lang=de (visited 2024-07-11)
[56] https://www.semanticscholar.org (visited 2024-07-11)
[57] Data retrieved from https://orkg.org on 2024-02-23.

encoded scientific contributions was employed in V and VI. Unsurprisingly, the MinSKG but also the ORKG did not pose any size-related issues for the tasks performed on the SKGs in the context of RDFtex. These tasks include the retrieval of data from the SKGs and the integration of new data into the SKGs. The data to be integrated is contained in the RDF documents that are constructed by RDFtex's export functionality. Since these export documents comply to the RDF standard and signify real-world entities, they qualify as KG fragments. Each export document comprises all scientific contributions of a paper that have been marked up using the provided export commands in the LaTeX source code.

While RDFa demonstrates how documents based on XML can be enriched with RDF information, RDFtex's export functionality goes one step further by applying similar ideas in the context of less structured LaTeX documents. Others have also acknowledged the need for such a functionality in order to facilitate the shift towards KG augmented research. One example is SciKGTeX (Bless et al., 2023), a package for semantically annotating contributions in scientific publications, thereby facilitating the integration of contributions into SKGs. But, to unlock the full potential of a publication form tailored to KG augmented research, a purely integration, i.e., export, oriented perspective is insufficient. Hence, RDFtex also introduces the novel idea of directly leveraging the rich information within SKGs through its complementary import functionality. This extends the opportunities that arise from the utilization of SKGs: Using RDFtex, researchers cannot only interact with SKGs through additional user interfaces to retrieve and integrate scientific information but also through research publications directly. Considering the high rigidity of the prevalent research and publication process as well as the LaTeX ecosystem with its long tradition, this result is particularly interesting in the context of this thesis: The application demonstrates that RDF technology can

- be employed to add semantic markup even to data with little structure,
- be added to systems retrospectively with only a small impact on existing processes, and
- be embedded into build pipelines with only a small impact on the pipeline runtime.

Of course, there are also challenges that surfaced during the investigation of this application: Despite its simple data model, users require considerable support when they have to directly interact with RDF data. In this regard, one problem that quickly becomes apparent is that the inclusion of lengthy IRIs deteriorates the readability of LaTeX source code. While RDFtex's import functionality only requires a single command and IRI per import, the export functionality uses multiple commands per export that comprise different IRIs. Even with relatively few imports and exports per document, the number of IRIs accumulates fast. As a result, working on LaTeX source files becomes increasingly tedious. Presumably, this not only applies to LaTeX source code used in combination with RDFtex or SciKGTeX but virtually any type of source code that is enriched with RDF data. In fact, XML documents that make use of RDFa exhibit similar problems, as Listing 2 indicates. However, the difference is that XML documents are not intended to be manipulated manually as often as LaTeX source files. When the recurrent manipulation of RDF-enriched source code cannot be avoided, the design of an efficient syntax has to be highly prioritized. This includes the consideration of mechanisms such as the prefix syntax featured in RDFa and other RDF technologies. RDFtex was designed with this particular feature up front.

The identification of relevant IRIs in raw RDF data was also found to be challenging. Serialization formats that focus on readability such as Turtle cannot mitigate this problem, as the user study showed: In order to investigate the interaction between users and raw RDF data, the deliberate choice was made to give the test subjects access to the MinSKG only as a Turtle document. Even users at the research associate level have not been able to identify the IRIs in the Turtle document that are necessary to solve the given tasks quickly. Hence, end-users should not be confronted with raw RDF data disregarding the employed serialization format. Instead, end-users should only be provided more abstract views on the data, ideally tailored to a user's current objective. In the context of RDFtex, [VI] discusses different tooling ideas facilitating the identification of IRIs in SKGs. One of them is the extension of academic search engines, as already mentioned above.

5 Quality Assessment of Software Repositories (Application 3)

The final application to be discussed in this thesis resides in the field of mining software repositories (Kagdi et al., 2007). In the intricate landscapes of contemporary business and academia, software repositories stand as central hubs for knowledge and innovation. In this regard, Git repositories in combination with sophisticated and powerful platforms such as GitHub[58] and GitLab[59] emerged as the de facto standard platforms for the collaborative development and distribution of source code and various other types of media vital for the functioning and progress of organizations and research initiatives alike. Due to the diverse quality criteria that individual repositories require, the management of repositories quickly becomes a time consuming chore as their number rises.

To give an example, reproducibility represents a multi-faceted aspect associated with the quality of repositories. The pursuit of reproducibility is fundamental in both scientific research and industrial development. The ability to replicate and validate findings, algorithms, and experiments hinges upon the reliability of the underlying software infrastructure. Yet, achieving reproducibility requires more than mere compliance with generic standards. Instead it demands the implementation of quality criteria tailored to this objective.

The application discussed in the following encompasses the automated validation of GitHub repositories against sets of predefined quality criteria that align with the requirements of the respective project. For this purpose, additional RDF technologies are employed that facilitate the efficient assessment of KGs against constraints representing these quality criteria.

5.1 Content and Positioning of Paper VII

The next content is contained in [VII] (Martin & Henrich, 2022b). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information and details on my contributions to this paper can be found in the back of this thesis.

5.1.1 Key Contributions

There are different approaches for implementing the automated validation of GitHub repositories against quality criteria. Due to the RDF ecosystem's representational capabilities, an approach was chosen that builds upon three components:

1. A representation of the repository that encompasses the relevant repository (meta)data.
2. A representation of the quality criteria.
3. A tool capable of evaluating repository representations against quality criteria representations.

[58] https://github.com (visited 2024-11-26)
[59] https://gitlab.com (visited 2024-11-26)

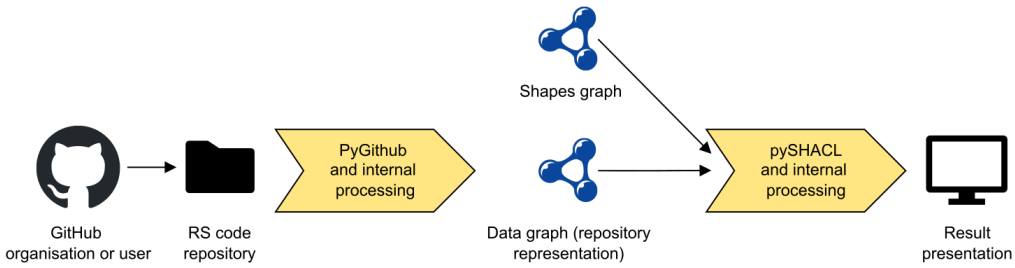


Figure 13: A simplified illustration of QuaRe's validation process with focus on the generation of representations of the repositories of interest, which are called data graphs in the SHACL context. Figure adopted from [VIII].

In other words, an application implementing this approach requires the ability to make statements about other statements. However, RDF can do this only through cumbersome workarounds such as reification (Hayes & Patel-Schneider, 2014). Conveniently, the RDF ecosystem provides multiple alternatives to tackle this problem. Paper [VII] discusses an approach based on the Shapes Constraint Language (SHACL) (Knublauch & Kontokostas, 2017) but also considers another approach based on OWL. Moreover, it introduces a tool called QuaRe implementing these two approaches. The current version of QuaRe does, however, not include the OWL approach anymore as it has been deprecated because of significant disadvantages, which are addressed in the following. Being implemented as a web application, users can access QuaRe through their browser. The application consists of two views, a validation page and a specification page. The discussion of the user interface is postponed to the section on [VIII] since this paper introduces significant usability improvements.

To model constraints for RDF graphs, which are called data graphs in the SHACL context, SHACL uses shapes graphs, i.e., RDF graphs with special vocabularies. To validate data graphs, SHACL validators like pySHACL[60] are employed to check whether the relationship structures and entities within data graphs comply with the constraints defined in the shapes graphs. If violations are encountered, comprehensive reports with details on the violations are automatically generated by the validators. Transferred to the application under consideration here, the GitHub API is accessed via the PyGithub wrapper library[61] to retrieve the relevant (meta)data about the repositories of interest. This data is then used to generate individual KG fragments that appropriately reflect the repositories and serve as the data graphs. Figure 13 illustrates this process.

In addition, shapes graphs that reflect the quality criteria are devised for each project type. To give an example, Figure 14 shows a simple shapes and a simple data graph. In this particular case, the validation would fail because the repository representation on the right side reveals that the repository contains two branches while the shapes graph prescribes that a finished research project must comprise exactly one branch. The presentation of the full ontology underlying the repository representations and the shapes graph is also postponed to the section on [VIII], as this paper introduces significant improvements to these artifacts.

[60] https://github.com/RDFLib/pySHACL (visited 2024-08-12)
[61] https://github.com/PyGithub/PyGithub (visited 2024-08-12)

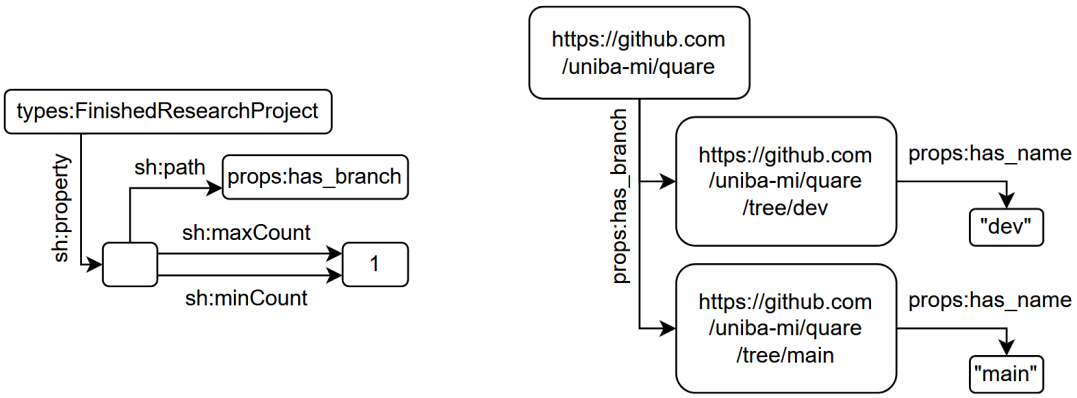


Figure 14: Left: A part of a shapes graph defining the quality criterion that a repository containing a finished research project must possess exactly one branch. Right: A part of a data graph derived from a GitHub repository. Figures adopted from [VII].

In comparison, the OWL approach detects violations based on the satisfiability of class expressions: project types are modelled as classes that come with class expressions for imposing certain restrictions on the properties of classes, reflecting the quality criteria. Using the (meta)data of the target repositories retrieved from the GitHub API, individuals of these classes are created whose properties reflect this data appropriately. Under the closed world assumption, a semantic reasoner can then infer the satisfiability of the class expressions, i.e., validate whether the properties of the individuals contradict the class expressions of the associated classes. However, the OWL approach has two disadvantages that justify why this option has been ruled out in favor of the SHACL approach:

1. Semantic reasoners such as Pellet terminate as soon as the first violation is encountered. Transferred to the application examined here, this would mean that users will only receive reports that comprise the first encountered violation. In comparison, SHACL validators do not terminate early and provide a report of all present violations, which is preferable from the user perspective.

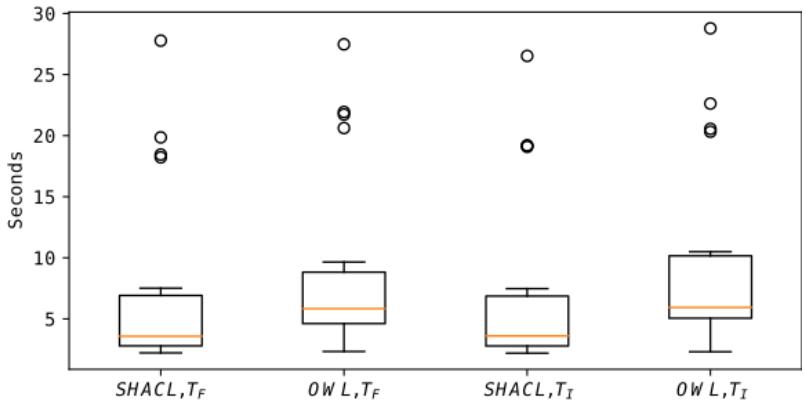


Figure 15: The results of the initial runtime evaluation of QuaRe. Apart from the employed approach, the x-axis mentions the examined project types: T_F is a project type comprising eleven quality criteria and T_I another one comprising four quality criteria. Figure adopted from [VII].

2. While SHACL has been specifically designed for the specification and validation of constraints, OWL possesses a wide range of logical capabilities. This results in a performance penalty, even if only a fraction of these capabilities are leveraged for the application discussed here. Accordingly, Figure 15 from [VII] shows that the SHACL approach is faster than the OWL approach disregarding the number of quality constraints constituting the examined project types.

5.1.2 Positioning of the Paper

Today, software repositories are very complex in terms of their content. They comprise source files, configuration files, documentation, project management functionalities, and many more. In terms of opportunities, Application 2 showcases how the flexibility of RDF permits the creation of representations of artifacts that are tailored to the specific requirements of various project types. Its small memory footprint and simple data model facilitate the ad-hoc generation of low-profile KG fragments, which is particularly advantageous in cases where the efficient retrieval of application-specific information is essential. The shapes graphs can also be easily extended to cover other project types and quality criteria, as [VIII] shows. Nevertheless, it is recommendable to anticipate conceivable future extensions when designing data graphs that are to be validated against shapes graphs. To give an example, Figure 16 shows another option for testing the quality criterion from Figure 14. In this case, the data graph explicitly states whether exactly one branch is present or not via a boolean value, making the validation of this criterion trivial. However, future project types could call for another cardinality of branches such as *at least two*, which cannot be validated based on the modified representation even though such a cardinality could be a conceivable criterion, e.g., for active software repositories.

In her bachelor thesis, which was supervised by me, Leoni Pflaum (Pflaum, 2022) also uses RDF to generate representations of software repositories, in her case, to browse the characteristics of software repositories. By means of the tool she implemented, users can, for example, identify developers with a certain expertise based on their contributions to previous projects. Accordingly, the structure of her KG fragments differs from QuaRe's to allow for the efficient retrieval of the necessary information. Additionally, her implementation operates on GitLab repositories instead of GitHub repositories.

Making statements about statements is a common requirement in many applications. Given the RDF context, it is crucial to carefully select the appropriate solution the ecosystem offers. While OWL is a tempting option due to its strong logical foundation and expressive power, it may not always be the

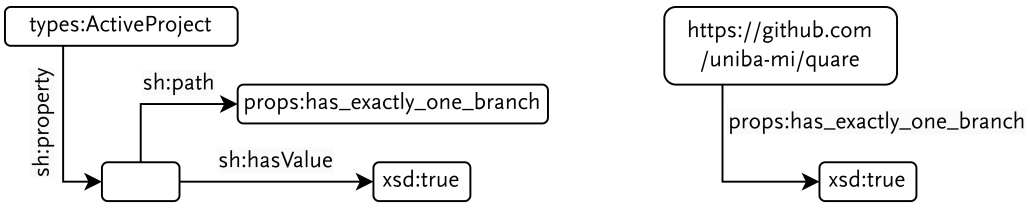


Figure 16: A modified version of the graphs from Figure 14. Left: A part of a shapes graph defining the quality criterion that a repository containing a finished research project must possess exactly one branch. Right: A part of a data graph derived from a GitHub repository.

best choice for every logical problem due to its complexity and resource demands. For scenarios requiring more streamlined and efficient approaches, alternative technologies like SHACL offer significant advantages. SHACL provides a concise scope for defining and validating graph shapes, enhancing performance without compromising the expressiveness needed for many practical applications. QuaRe leverages SHACL to ensure that repositories meet predefined standards, thereby facilitating better code quality and consistency. By utilizing SHACL for the validation, the application can efficiently check various aspects of the repositories and adherence to best practices, providing immediate feedback and ensuring that quality criteria are consistently met.

5.2 Content and Positioning of Paper VIII

The next content is contained in VIII (Hummel et al., 2024). In the following, the paper's contributions relevant for this thesis are briefly summarized. Refer to the original publication for detailed information. After the summary, the paper is positioned in the context of this thesis. Bibliographic information and details on my contributions to this paper can be found in the back of this thesis.

5.2.1 Key Contributions

Even though the project types discussed in VII comprise realistic quality criteria, they still mainly serve as a proof-of-concept. To show QuaRe's capabilities in practice, VIII shifts the focus towards a highly relevant topic: different organizations suggest the FAIR principles, which require data (and by extension also software) to be findable, accessible, interoperable, and reusable (Hasselbring et al., 2020), as a countermeasure for the still ongoing reproducibility crisis in research (Deutsche Forschungsgemeinschaft, 2022). As a contribution to the FAIRness movement, VIII introduces a new project type whose quality criteria are derived from the FAIR principles, thus facilitating the evaluation of the FAIRness of software repositories. Additionally, the paper discusses improvements applied to QuaRe regarding the generated repository representations and the application's general usability.

Focussing on the new project type called *FAIRSoftware* first, Figure 17 shows the fragment of the shapes graph specifying the associated quality criteria. Each of the ten node and property shapes directly attached to the FAIRsoftware node reflects one quality criterion for FAIR software that has been derived through literature analysis. In cases where multiple options fulfill a criterion or where multiple aspects have to be correct to fulfill a criterion, the logical operators of SHACL are employed.

To be able to test a repository against this graph, the repository representation, i.e., the data graph, has to be modelled adequately. As shown in Figure 18, the majority of the employed properties originates from the vocabulary of the Software Description Ontology[62] but custom properties are used in cases where no suitable properties could be found. To maximize performance, only the necessary repository information is retrieved from GitHub, which is why a repository representation does not include information about issues when it is evaluated against the *FAIRSoftware* project type (cf. Figure 17).

[62] https://w3id.org/okn/o/sd (visited 2024-07-02)

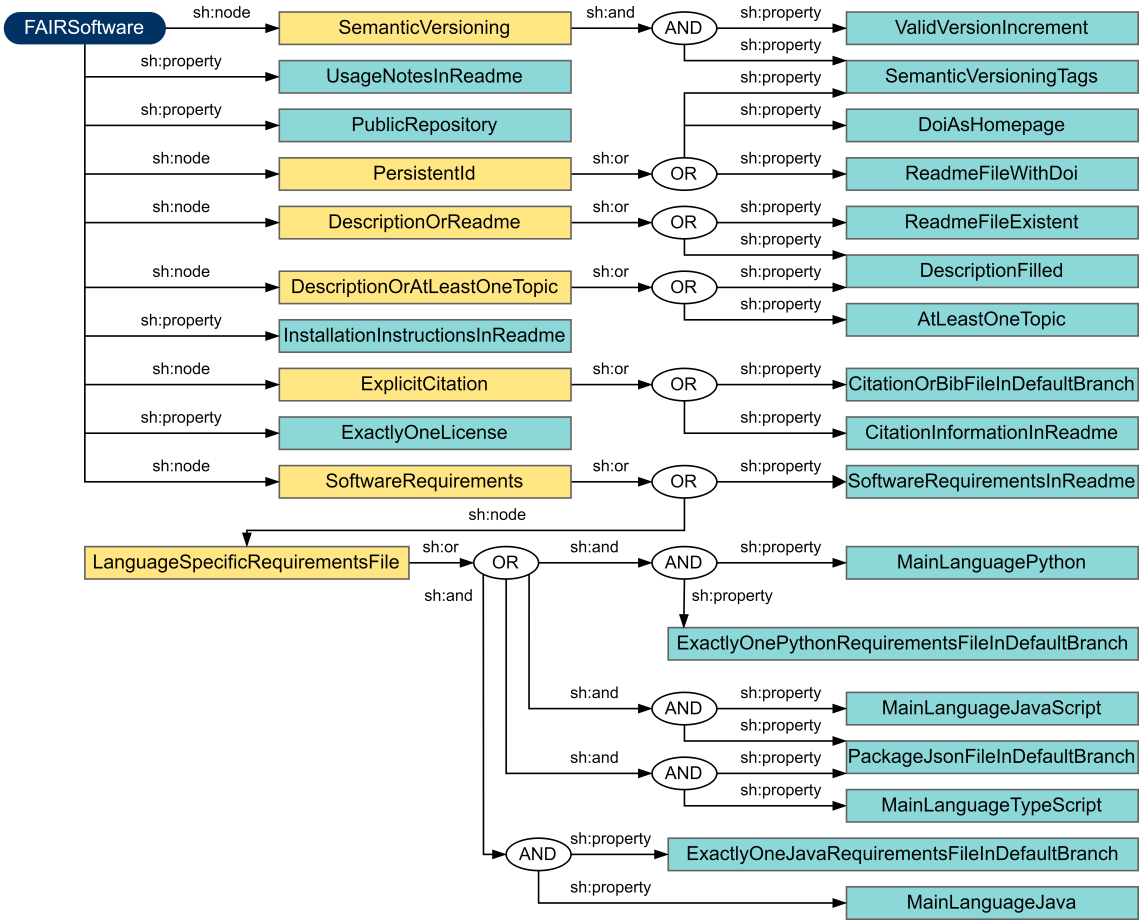


Figure 17: The fragment of the SHACL shapes graph representing the FAIRSoftware project type; other project types are omitted here. The project type shape for the FAIRSoftware project type is depicted in blue, the corresponding node and property shapes are light yellow and turquoise. *sh* refers to the SHACL namespace. Figure and caption adopted from [VIII].

The QuaRe web application features two views: a validation and a specification page. As shown in Figure 19, the validation page allows entering the names of an arbitrary number of GitHub repositories of interest and selecting the project types against which the repositories are to be validated. For validating a large number of repositories or performing multiple validation in short succession, users might be required to additionally enter a personal GitHub access token[63], which can be easily generated in the GitHub web interface. Once the information is added, users can click the submit button to issue the validation to the backend. When the backend has returned the validation results, an interface element displaying the share of fulfilled quality criteria and a button labeled *View* appear next to each repository project type combination. Clicking on this button reveals two additional text areas. The left one contains the raw explanation, i.e., the validation report as generated by the SHACL validator pySHACL[64] that

[63] https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens (visited 2024-07-08)
[64] https://github.com/RDFLib/pySHACL (visited 2024-07-08)

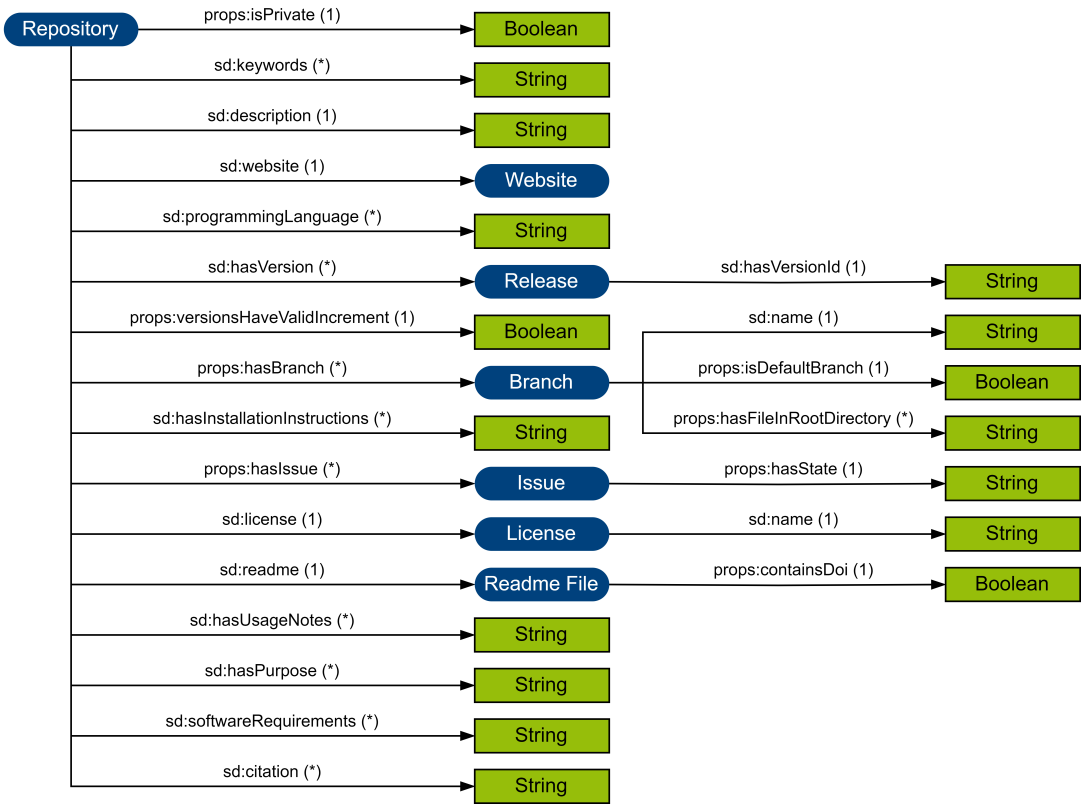


Figure 18: An abstract visualization of the ontology underlying the repository representations (data graphs). IRIs are depicted in blue, literals in green. The cardinality is given in brackets. *sd* refers to the Software Description Ontology namespace, and *props* to the namespace for additional custom properties. Figure and caption adopted from [VIII].

is employed in the backend. End-users that are not familiar with SHACL and RDF cannot be expected to comprehend raw SHACL reports, though. For this reason, the right text area presents a verbalized explanation that is more accessible. In QuaRe's current version, verbalized explanations are generated based on the values of the `sh:message` properties of the violated node and property shapes. When this property is added to the shapes graph and the shapes in question are violated, the property value is automatically included in the validation report through the property `sh:resultMessage` such that they can be extracted and presented in a more user-friendly manner (cf. verbalized explanation in Figure 19). However, messages provided this way have to be added to the shapes graph by hand which makes the addition of new shapes cumbersome. Therefore, experiments have been conducted exploring whether verbalized explanations can be generated by feeding the raw validation reports alongside appropriate prompts into LLMs. In comparison to Google Bard (now Google Gemini), ChatGPT managed to produce better results but the texts still occasionally contained hallucinations regarding recommended actions to resolve encountered violations, failed to mention all eligible options to fulfill quality criteria, or paraphrased shapes incorrectly.

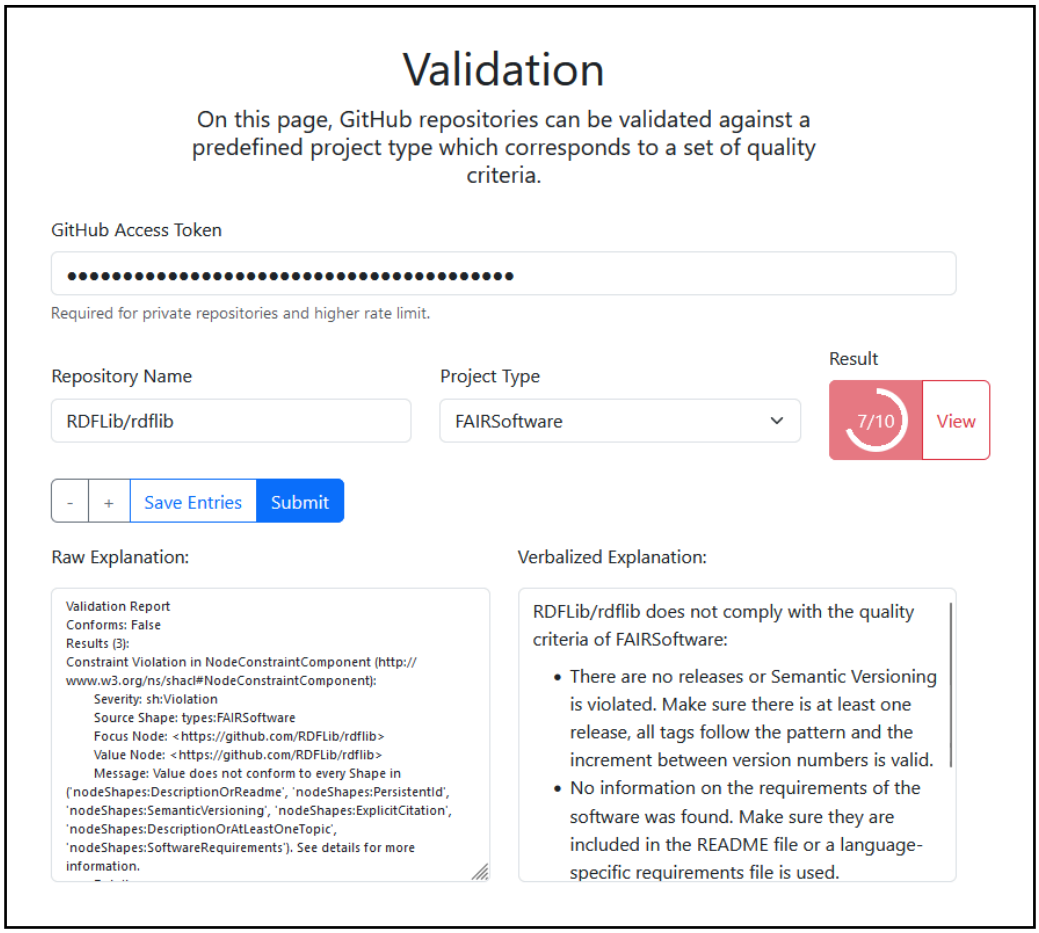
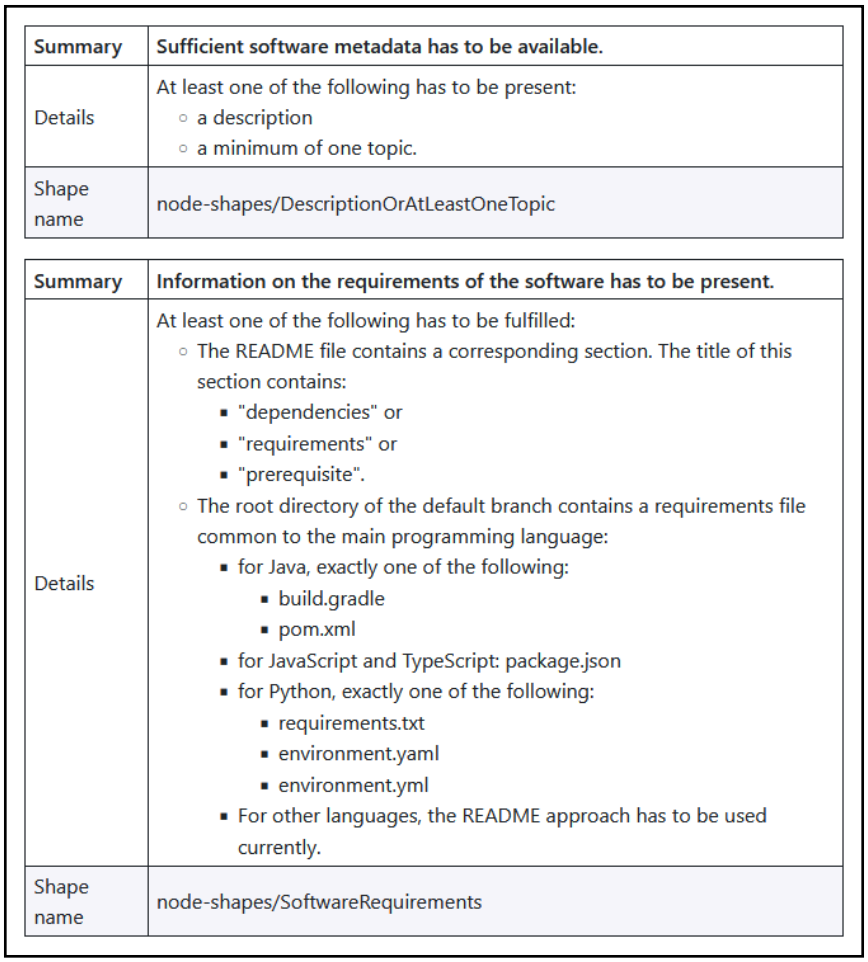


Figure 19: A screenshot of the validation page: In response to a click on the submit button, a repository has been validated against the new project type *FAIRSoftware*. Afterwards, the button labeled *View* was clicked, revealing the raw and verbalized explanations. Figure and caption adopted from [VIII].

Thus, more prompt engineering and iterative user evaluation are required before LLM technology can be employed in QuaRe in production. Complementing the validation page, QuaRe's specification page gives an overview of available project types and provides a user-friendly verbalized description of the associated quality criteria (cf. Figure 20). In this case, the verbalizations are stored as the values of the `sh:description` property, which again has been manually added to the shapes in shapes graph. Accordingly, this represents another opportunity where LLM technology could be used to automate the provision of information about shapes, which should be explored in the future.

5.2.2 Positioning of the Paper

The user study conducted in the context of Application 2 revealed that end-users should not be confronted with RDF data directly. The same applies to Application 3. Here, the employed SHACL validator generates violation reports that are not comprehensible without a profound understanding of RDF and



Summary	**Sufficient software metadata has to be available.**
Details	At least one of the following has to be present: ◦ a description ◦ a minimum of one topic.
Shape name	node-shapes/DescriptionOrAtLeastOneTopic

Summary	**Information on the requirements of the software has to be present.**
Details	At least one of the following has to be fulfilled: ◦ The README file contains a corresponding section. The title of this section contains: ▪ "dependencies" or ▪ "requirements" or ▪ "prerequisite". ◦ The root directory of the default branch contains a requirements file common to the main programming language: ▪ for Java, exactly one of the following: ▪ build.gradle ▪ pom.xml ▪ for JavaScript and TypeScript: package.json ▪ for Python, exactly one of the following: ▪ requirements.txt ▪ environment.yaml ▪ environment.yml ▪ For other languages, the README approach has to be used currently.
Shape name	node-shapes/SoftwareRequirements

Figure 20: A screenshot of QuaRe's specification page showing the specification of two quality criteria. Figure and caption adopted from [VIII].

of the role shapes graphs and data graphs play. Especially considering the extended usage scenario of QuaRe, this is problematic: As pointed out, QuaRe's primary purpose is to validate software repositories against sets of quality criteria. This way, users can identify which characteristics of repositories of interest violate, e.g., FAIR best practices. However, the usage scenario does not end there: When violations are identified, users should also receive guidance on how they can be resolved. While the comprehension of violations from the validation report is already difficult, the derivation of effective resolving actions is even harder. Similarly, the specification of project types and quality constraints presented on QuaRe's specification page must convey the relevant information in a user-friendly form. As an interim solution, [VIII] proposes the addition of manually written explanatory text to SHACL shapes via special properties that can be leveraged by QuaRe's frontend to make validation reports as well as project criteria specifications more accessible. In the future, these texts could be generated via LLMs to facilitate the addition of new quality criteria by users. As pointed out in [VIII], the first experiments in this regard revealed flaws but were still promising, again indicating the opportunities a combination of symbolic and subsymbolic technologies could provide.

Table 7: An example of how a best practice that is derived from an abstract FAIR principle is translated into a compound rule that can implemented in SHACL. Based on [VIII].

ID	Best Practice	SHACL-Compatible Description
BP10	Software requirements are available (from the FAIR principle *Reusable*)	There must be at least one section in the README file where the lower-cased title contains *dependencies*, *requirements*, or *prerequisite*. Alternatively, depending on the main programming language, precisely one file specifying the requirements must be present in the root directory of the default branch. Currently, this is limited to only four popular languages and a selection of admissible files: • For JavaScript and TypeScript: a *package.json* file • For Python: a *requirements.txt*, *environment.yaml* or *environment.yml* file • For Java: a *pom.xml* or *build.gradle* file

Paper [VIII] reveals another important insight related to the translation of abstract guidelines into concrete rules that can be validated via RDF and SHACL. Originally, the FAIR criteria have not been devised explicitly with software repositories in mind. Additionally, GitHub repositories are not compatible with RDF and SHACL out of the box. Hence there are three layers that come into play for the FAIR use case:

1. The original four FAIR principles that require research data to be findable, accessible, interoperable, and reusable.
2. An operationalization of the abstract FAIR principles that tailors them to research software provided through GitHub repositories. The operationalization used in [VIII] comprises ten best practices that have been composed based on previous work (Iglesias-Molina & Garijo, 2023).
3. An implementation of the best practices as a shapes graph that allows the validation of a suitable data graph representation of the repository against the ten best practices.

The establishment of these layers is not only necessary for the FAIR use case examined here but also for many others where abstract business guidelines are to be translated into shapes graphs such that they can be validated using QuaRe. As an example, consider Table 7, which shows how the best practice BP10 from (Iglesias-Molina & Garijo, 2023) has been interpreted such that it can be implemented as SHACL shapes. As a basis, the *Reusable* FAIR principle (Dumontier, 2022) states that (meta)data are richly described with a plurality of accurate and relevant attributes, which includes the provision of a data usage license, detailed information on the data provenance, the compliance with domain-relevant community standards. Applied to research software as a special kind of research data, Iglesias-Molina & Garijo (2023) derive, among others, the best practice that software requirements have to be available based on this principle. Implementing BP10 in the shapes graph, reveals that there are still significant decisions that have to be made since software requirements can be documented in a variety of ways. One option is to provide a textual description, which can be placed in various places in the README

file. Another option to use language-specific requirements files[65]. Since SHACL depends on specific rules, each option and variant that is to be supported has to be added explicitly. This shows the difficulty of translating abstract business guidelines into shapes graphs.

Application 3 is the final application examined in this thesis. The upcoming section thus proceeds to a summarizing discussion of the insights obtained through the investigation of the three applications with the goal of answering the three top-level research questions of this doctoral thesis.

[65] There are even more options that are used in practice. Additionally, note that the mentioned options are only effective for software repositories that comprise a single project, i.e., provide a single README file or a single requirements file.

6 Discussion of Insights

The three applications presented in the previous sections leverage various technologies from the RDF ecosystem and operate on KGs with varied scopes and sizes. This section revisits and summarizes the insights obtained from the investigation of these applications to derive opportunities, challenges and recommendations for the utilization of RDF-based KGs, thereby answering the top-level research questions 1 , 2 , and 3 of this doctoral thesis, which have been defined in Section 1. Recall that the goal is not to provide an exhaustive list of all opportunities, challenges, and recommendations that apply to the utilization of RDF-based KGs. Instead, the goal is to extend our understanding thereof based on the investigated applications. Hence, some of the paragraphs below elaborate on insights from previous work (cf. Section 2.2) while others add novel aspects to this line of research.

6.1 Opportunities

The first top-level research question to be discussed here, i.e., 1 , addresses the opportunities that arise from the utilization of RDF-based KGs for particular applications. This includes benefits that originate from the characteristics of the RDF ecosystem but also from the concept of KGs in general.

The opportunities a technology offers are the first reference point for potential users to think about using the technology. However, individuals might be inclined to employ a technology when its opportunities are praised exaggeratedly by different sources, despite the existence of significant challenges that accompany its utilization. Gartner's Hype Cycle shown in Figure 2 illustrates the negative impact that inflated expectations can cause. Consequently, it is essential to consider the opportunities pointed out in the following only in conjunction with the challenges delineated in the subsequent section. This approach facilitates forming a more comprehensive understanding, thereby reducing the likelihood of unrealistic expectations.

6.1.1 A Rich Ecosystem for the Management of Heterogeneous Data

The first opportunity to be discussed here has been commended since the initial proposal of RDF, namely that the framework has been explicitly designed to seamlessly integrate heterogeneous data, thereby addressing the complexities inherent in modern (distributed) data environments. For this purpose, RDF both allows storing data internally *and* referencing external sources:

1. To store data internally, i.e., within an RDF graph itself, IRI nodes are linked via properties to literal nodes, which can contain structured, semi-structured, and unstructured data. Literal nodes support a variety of XML Schema data types including strings, numbers, and dates, and even binary data through hexadecimal and base64 encoding (Cyganiak et al., 2014). By explicitly integrating data through literal nodes into KGs, external dependencies can be eliminated and abstract representations of resources with exactly the information required for specific application can be constructed.

2. Data can also be referenced rather than explicitly included in the RDF. In this case, IRI nodes function as pointers to external resources of arbitrary types. Through these IRIs, applications can access the external data for further processing. The provision of such links facilitates the interconnection of separate systems, allowing to share and reuse data without centralization[66]. In place of the more generic IRIs, URLs are typically employed for the nodes such that the external data can be accessed more easily through standard protocols such as HTTP.

The applications examined in this thesis leverage KGs that use both options. To give some examples, Wikidata from Application 1 manages a large amount of information on the comprised entities and properties as internal data but also provides identifiers from other knowledge bases, databases, encyclopedias etc. that describe identical entities. For instance, for the entity *Earth* (Q2) there are the following triples in Wikidata, among others:

Earth (Q2) —exact match (P2888)→ https://dbpedia.org/page/Earth

Earth (Q2) —GND ID (P227)→ https://d-nb.info/gnd/1135962553

Earth (Q2) —BabelNet ID (P2581)→ https://babelnet.org/synset?id=bn:00029424n&lang=EN

The first triple links the Wikidata entity to its DBPedia entry, the second to its entry in the international authority file of names, subjects, and organizations by the German National Library called GND[67], and the third to its BabelNet[68] entry. The MinSKG from Application 2 uses IRIs to reference external resources such as images, datasets, and software. In contrast, the raw text that is marked up using the RDFtex export commands is directly included in the MinSKG at the end of the export process. Theoretically, images, for example, could also be integrated directly using base64 encoding. In the end, KG maintainers have to decide which data is included internally and which data is referenced externally.

The applications examined in this thesis show that its ability to integrate and manage heterogeneous data is only one benefit the RDF ecosystem offers: The applications differ not only with respect to the scopes and sizes of employed KGs but also serve entirely different use cases from distinct application domains. The richness of the RDF ecosystem, which is a result of the broad array of standards and technologies such as SPARQL for querying, various serialization formats like Turtle for data interchange, SHACL for validating RDF data shapes, and OWL for reasoning on complex ontologies, enables the implementation of applications with diverse and demanding requirements. These technologies, all inherent to the RDF ecosystem, support various tasks related to KGs within a unified framework. Consequently, developers familiar with the ecosystem are able to implement applications that provide a wide range of functionalities entirely with technology from the ecosystem, i.e., with only little use of additional external technology.

[66] Without such links, error-prone entity linking techniques would be required to retrieve information on the same entities and properties from different sources.
[67] https://www.dnb.de/EN/Professionell/Standardisierung/GND/gnd_node.html (visited 2024-07-20)
[68] https://babelnet.org/about (visited 2024-07-20)

6.1.2 Knowledge Graphs as a Lean On-Top Technology

From an organization's perspective, it is vital to make educated decisions based on the knowledge at hand, which is potentially spread across a range of heterogeneous resources. For this purpose, the notion of Knowledge Management Systems (KMSs) emerged, which refers to systems that facilitate knowledge creation, knowledge storage and retrieval, knowledge transfer, and knowledge application processes in organizations (Alavi & Leidner, 2001). To discuss the way KGs ingest and provide knowledge, the following paragraphs will compare them to data lakes. Data lakes are a state-of-the-art option for handling and processing large volumes of heterogeneous data (Mathis, 2017) and also capable of implementing KMSs. They ingest data following the *schema-on-read* principle[69], which states that data is stored as is and transformed into required formats only on demand. In return, this flexibility necessitates sophisticated management and governance processes. To be able to store heterogenous data as is, data lakes integrate various data storage options such as relational databases but also key value stores, document stores, and even graph databases like Neo4J (Mathis, 2017).

In contrast, KGs take a different approach for integrating heterogeneous data: KGs do not store resources of interest within them as is but instead rely on the assignment of IRIs to these resources in order to express relevant statements about them. Figure 21 illustrates this concept by means of an example use case where product descriptions and images have to be extracted from an e-commerce platform. This approach has two important implications:

1. Only information that is explicitly added becomes available through KGs.
2. KGs are comparatively easy to setup and operate in relation to their high utility.

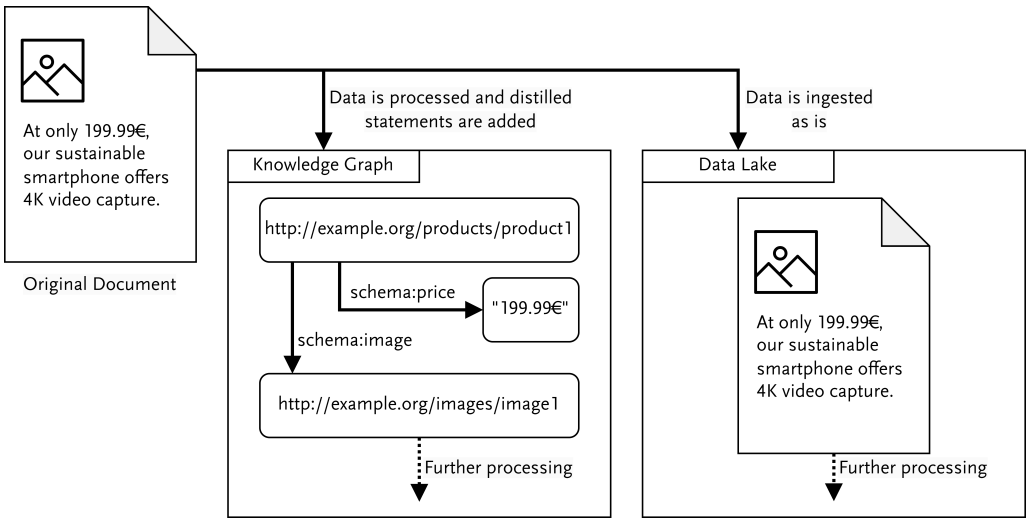


Figure 21: A simplified illustration of how KGs and data lakes integrate data. The example use case is to ingest a document from an e-commerce platform to extract prices and pictures of products. The shown KG uses properties from the schema.org vocabulary.

[69] Note that the term schema in schema-on-read is unrelated to RDFS and RDF vocabularies. In the context of data lakes, the term schema refers to a template that specifies the format of some data for a certain purpose.

Intuitively, the first point might appear like a major downside but in practice many applications only require a subset of the available information encoded in resources anyway. In fact, one can view this point as an advantage, as well: Without sophisticated governance mechanisms, storing large volumes of data as is, like data lakes do, renders them susceptible to degenerating into data swamps that exhibit uncertainty about the origin and reliability of data as well as about appropriate data protection (Chessell et al., 2014), thereby heavily impacting their utility, also because of legal implications, e.g., regarding data privacy (Mathis, 2017). In the case of KGs, it is easier to retain the overview of what data is actually present since composing statements and adding them to KGs is an explicit process that can be closely monitored. In Figure 21, for instance, the KG only contains the information that is required for the use case at hand. In contrast, the data lake integrates the original resource itself, which allows serving other use cases in the future. In this example, the data lake would also allow extracting information about product features. This information is not available in the distilled KG representation. At the same time, data lakes are more expensive in terms of system resources and due to their dependence on more sophisticated data governance processes. Nevertheless, proper data governance is still a pivotal requirement for KGs, as well, which will be addressed later.

Ultimately, data lakes are a powerful tool for data discovery, analytics, ad-hoc investigations, and reporting (Chessell et al., 2014). At the same time, they come with significant complexity and require substantial effort to set up and maintain effectively. In comparison, KGs are lean, which facilitates their usage as an on-top technology that can be added to existing systems retrospectively with relative ease. In particular, Application 3 leverages this capability, where the QuaRe tool serves as an add-on to the monolithic GitHub platform, providing access to an abstracted representation of software repositories. It is conceivable to transfer this approach to other cases in which leveraging concise abstractions of larger resources could prove advantageous. One potential advantage is the increased efficiency of the processing of these abstractions. Another potential advantage is that these abstractions are RDF-compatible, which enables the usage of the many technologies within the RDF ecosystem on the data.

Application 3 generates the repository representations ad-hoc and on demand. Following the data lake terminology, one could thus informally call Application 3's approach *RDF-on-read*. On the contrary, in other applications, it might be desirable or necessary to generate and store such RDF representations before their actual use. In reminiscence of the *schema-on-write* principle (Mathis, 2017), which states that data is converted into a desired schema and stored in advance, one could informally call this approach *RDF-on-write*. One advantage of *RDF-on-read* would be that the abstractions always reflect the newest data. However, when the amount of data exceeds a certain threshold, preparing the abstractions beforehand via *RDF-on-read* might be the only feasible option. In any case, in terms of opportunities, it is commendable that both *RDF-on-read* and *RDF-on-write* are possible.

In summary, KGs enhance the ability to connect and query heterogeneous data, making them a versatile and practical choice for organizations looking to extend their data infrastructure with a low impact on existing systems. Especially when high-quality data is available, KGs can be built without extensive restructuring of data.

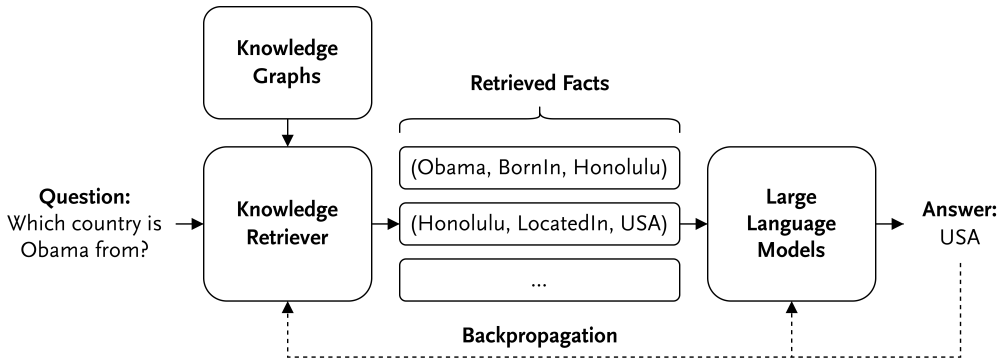


Figure 22: A simplified illustration of how the RAG models retrieve external knowledge to enhance the LLM generation. Figure and caption adapted from Pan et al. (2023).

6.1.3 A Promising Future

Although RDF technologies like RDFa as of today have not achieved the widespread adoption anticipated by its proponents, the KG community remains active and committed. The increasing number of research papers published each year, as documented in dblp's record (cf. Figure 1), highlights the community's ongoing dedication and engagement. These publications cover a broad range of topics, from theoretical advancements to practical applications, demonstrating the technologies relevance. By employing KGs with varied scopes and sizes, the applications examined in this doctoral thesis give another example of the diverse applications for which KGs can be effectively employed.

New research directions and applications for KGs continually emerge, reflecting the field's dynamic nature. One cutting-edge research area focuses on the combination of symbolic and subsymbolic machine learning methods. In this regard, one promising approach is neuro-symbolic AI (Hitzler & Sarker, 2021), which aims to exploit the structured information as provided by KGs to improve the performance and reliability of machine learning models. Due to high public attention, current research in this context is interested in eliminating the flaws of LLMs such as hallucinations and the lacking domain-specific knowledge (Pan et al., 2023). To give an example, an important technique called RAG (Lewis et al., 2020) tries to improve question answering by retrieving relevant facts, i.e., triples, from KGs, which are then fed into LLMs to generate the answer. Figure 22 illustrates this process. Qualitative investigation shows that the exploitation of symbolic facts causes RAG models to hallucinate less and to generate factually correct text more often than other models (Lewis et al., 2020).

In this thesis, the Applications 1 and 3 exhibit points of contact with the boundary between the symbolic and the subsymbolic domain. In Application 1, paths between entities in Wikidata, i.e., symbolic information, could be verbalized using subsymbolic LLMs to generate grounded textual explanations of the relationship between entities. Initial experiments in this regard achieved promising results. Application 3 provides a similar opportunity where the generated SHACL validation reports could be verbalized to provide users guidance on how to resolve the encountered violations. A thorough investigation of this idea, however, is left open to future work.

Apart from recent research, another indicator for the promising future of KGs is that many major corporations such as Microsoft, Google, Facebook, IBM, and others have successfully integrated KGs into their operations, using them to enhance search results, improve recommender systems, manage data more effectively, and for various other tasks (Noy et al., 2019). Google's Knowledge Graph powers its search engine to provide more relevant search results (Singhal, 2012), while Amazon utilizes their KG-based COSMO system to enhance its product recommendations (Yu et al., 2024). IBM uses its KG framework to power Watson Discovery, a system for knowledge discovery based on structured and unstructured data, but also to provide clients the necessary infrastructure to build their own KG on top of IBM's prebuilt KG (Noy et al., 2019). Another observation is that providers of cloud services continue building KG environments. One notable example that has already been mentioned in Section 2 is Amazon Neptune, a graph database part of Amazon Web Services that supports KGs based on RDF and on property graphs[70], more specifically Apache TinkerPop[71]. To make the migration to their cloud platform more attractive, Amazon Neptune supports different query languages:

- SPARQL for querying RDF-based KGs
- Gremlin[72], the query language that comes with Apache TinkerPop
- openCypher[73], the open-source version of the Cypher query language used in the previously mentioned Neo4J graph database[74]

The active development of Amazon Neptune and other KG infrastructure, in parts even open source, shows the industry's and the community's commitment to KG technologies. Thus, from an enterprise perspective, it is now easier and less risky than ever to engage with KG technologies and employ them to implement new products and services.

In accordance with the current positioning of KGs on Gartner's Hype Cycle, their full potential is still not known. Hence, there are numerous unexplored opportunities and applications to be expected in the future. Organizations that invest in and develop advanced KGs today could gain a significant competitive advantage as new machine learning approaches become viable for production use. As the technology matures, the ability to effectively utilize KGs will potentially become a key differentiator in various domains, especially considering the recent results achieved using neuro-symbolic AI.

6.2 Challenges

As said before, considering the challenges that accompany the utilization of a technology is mandatory for avoiding the risk of unrealistic expectations. Hence, the following sections cover relevant aspects that surfaced during the investigation of the three examined applications, thereby addressing the top-level research question 2.

[70] https://aws.amazon.com/de/neptune (visited 2024-07-11)
[71] https://tinkerpop.apache.org (visited 2024-07-15)
[72] https://tinkerpop.apache.org/gremlin.html (visited 2024-07-15)
[73] http://opencypher.org (visited 2024-07-15)
[74] https://neo4j.com/product/neo4j-graph-database (visited 2024-07-15)

Table 8: Size and order statistics of the KGs Wikidata, DBpedia, Freebase, and YAGO. Numbers adopted from Färber et al. (2018).

	Wikidata	DBpedia	Freebase	YAGO
Size (number of triples)	748,530,833	411,885,960	3,124,791,156	1,001,461,792
Order (number of entities)	18,697,897	4,298,433	49,947,799	5,130,031

6.2.1 Availability of Knowledge Graphs

The first challenge covered in this section is a problem well-known in many domains: the availability of resources. Applications that rely on certain data to provide some functionality require access to this data. When data of insufficient quality or with a different focus is leveraged, the applications' performance suffers. When no appropriate data is available, the applications' implementation might not be feasible at all. At the same time, the creation of datasets that fulfill the application-specific requirements can be expensive. Representing a special kind of dataset, this also holds for KGs. While the KG fragments used in Application 2 and 3 are comparatively unproblematic in this regard since they are generated ad-hoc on readily available high-quality data, the functioning of Application 1 and 2 heavily depends on the public availability of Wikidata and the ORKG or at least comparable alternatives.

More specifically, Application 1 demands access to an open-domain KG that is well-maintained and features a vast amount of entities and properties. If Wikidata would not exist, there are only a few alternatives including the previously mentioned DBPedia, Freebase, and YAGO. Table 8 provides statistics of these KGs, revealing that Freebase and YAGO actually exceed Wikidata in terms of size and, in case of Freebase, also in terms of order. Unfortunately, both Freebase and YAGO are no longer actively maintained, impeding their utility for Application 1. For most organizations, it is virtually impossible to devise KGs of this size. Additionally, the return of investment might not be estimable upfront, making the decision in favor of such a project even more difficult. Due to the domain-specific scope of the ORKG, which is employed in Application 2, the creation of a similar KG is easier but still requires significant effort. In this case, parsing research publication that leverage a variety of templates and are provided through a variety of publisher interfaces, in parts behind pay walls, is a significant challenge.

In this context, another aspect is also important to note: Just because a KG of an appropriate size and scope exists, does not automatically mean that it is also available. In the past, some KGs have been created as part of research projects but have ultimately not been published or are not available anymore. One example is KnowLife, a domain-specific KG for Health and Life Sciences, whose data is currently nowhere to be found, not even through the official web page[75]. And even if the data is available through interfaces, it still does not automatically mean that a KG can be employed for an application. For instance, Section 3 explained that the Wikidata Query Service is not able to return all adjacent entities for a significant number of entities, which is why I and II had to resort to a workaround using only the outgoing edges for the implementation of BiPaSs.

[75] https://knowlife.mpi-inf.mpg.de (visited 2024-07-19)

6.2.2 Data Quality Requirements

Strictly following the definition from (Hogan et al., 2021), an RDF graph only qualifies as a KG when its entities signify real-world knowledge. Based on this definition, it is actually debatable whether Figure 5 and Figure 6 from Section 2 show a KG or not. If the entities with the labels Bob and Alice refer to real-world entities, regardless of whether they are fictitious or not, one could call the examples KGs. In contrast, if the entities are mere placeholders, the graphs could be viewed as RDF graphs. At the same time, one could argue that even mere placeholders already sufficiently represent (fictitious) entities. Without additional information, only the authors of the RDF data can know for sure. This shows that there is room for interpretation regarding the boundary between RDF graphs and KGs[76].

Regarding the utilization of KGs, the information in KGs is faithfully considered true (Cyganiak et al., 2014) and a system can, for example, discover new knowledge using knowledge graph completion techniques (Chen et al., 2020) and reasoning (Hitzler et al., 2012), which from the system's point of view is also considered true. In other words, the system is indifferent as to whether the graph is a KG or just an RDF graph. That being said, algorithms that are tailored to operate on real-world data like the pathfinding algorithm of Application 1 that employs word embeddings trained on natural language text would perform bad when non-real-world entities are encountered. Since a KG is not able to identify incorrect or contrived data as long as it is consistent, it is the obligation of KG maintainers to make sure that added information complies with data quality requirements. Accordingly, maintainers should provide transparent information about the included data and implement appropriate data governance and quality assurance processes. Otherwise, the performance of applications might deteriorate and the KG could end up in a data swamp like state (Mathis, 2017).

At first glance, the KGs employed in the applications of this thesis are either well-maintained or generated single-handedly such that data quality is not a major issue. However, both Wikidata and the ORKG are quite liberal regarding the addition of new data and the manipulation of existing data. In the case of Wikidata, users are even able to edit all pages that have not been explicitly protected without creating a user profile beforehand[77]. The ORKG at least requires users to log in before they are allowed to edit data. Nevertheless, in both cases incorrect or contrived data can end up in the KGs relatively easy. Fixing this data is a task mainly left open to the respective user communities. Therefore, practitioners have to make sure that their algorithms are robust enough such that erroneous data does not impede their applications' performance. Clearly, some applications are able to deal with such data better than others: Due to the vast size and order of Wikidata, a large number of erroneous entities and properties would be required to significantly impede the quality of the found paths in Application 1. In Application 2, though, if a wrong dataset URL is added to a research contribution in the SKG, this URL will end up in the compiled PDF when the contribution is imported, which is a crucial problem considering the application domain. In summary, the well-known colloquial phrase *garbage in – garbage out*, which expresses the cause-effect relationship between poor input data and unreliable data output, also applies to many KG applications.

[76] The same also applies to KGs implemented using data models other than RDF.
[77] Like Wikipedia, Wikidata adds the user's IP as the author to the edit when the user is not logged in via a profile.

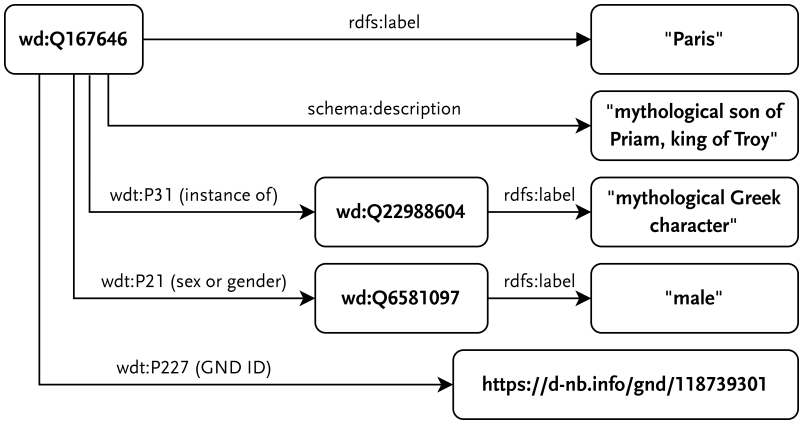


Figure 23: A selection of the information offered by Wikidata on the entity *Paris* (Q167646). The prefix rdfs is short for http://www.w3.org/2000/01/rdf-schema#, schema for http://schema.org/ and wdt for http://www.wikidata.org/prop/direct/. Data retrieved from https://www.wikidata.org/wiki/Q167646 on 2024-07-18.

Entity ambiguity, a well-known challenge in natural language processing, is related to this aspect. As presented in I, Application 1 originally relied only on entity labels to position entities in the priority queue. This naive approach is problematic because large open-domain KGs comprise a significant number of entities with similar or even identical labels, resulting in a compromised prioritization of candidate entities. Additionally taking the entity descriptions into consideration, as explained in II, to effectively disambiguate entities was an important change to improve the quality of the found paths. This shows that a KG, which can be utilized for a multitude of applications, both anticipated and unforeseen, is distinguished by its capacity to offer a variety of supplementary information for the entities it comprises. In Table 5, the entities *Paris* (Q167646), the mythological son of Priam, king of Troy, and *Paris* (Q90), the capital city of France, were used to illustrate the positive effect of the additional consideration of entity descriptions. Figure 23 shows that Wikidata, being a well-maintained KG, would even offer many other information that could be used to disambiguate *Paris* (Q167646) and *Paris* (Q90). As depicted, this includes taxonomic and other descriptive information as well as links to other knowledge bases where even more information could be retrieved.

6.2.3 Complexity of the RDF Ecosystem

The large number of standards and technologies mentioned and employed in this doctoral thesis gives on idea of the extent of the RDF ecosystem. However, there is still a significant number left. Some of them have been deprecated and replaced, while others simply have not been covered as they are beyond the scope of this thesis. A few examples for both categories are:

- Examples of deprecated or replaced technologies and standards:
  - RDQL (Seaborne, 2004), an alternative query language for RDF data, eventually surpassed by SPARQL as the standard RDF query language

  - DAML+OIL, an alternative web ontology language that heavily influenced OWL, the current standard web ontology language (Connolly et al., 2001)
- Examples of technologies and standards beyond the scope of this thesis:
  - HDT, a binary RDF serialization format for the efficient exchange of large volumes of RDF data (Fernández et al., 2013)
  - Hydra, a lightweight vocabulary facilitating the creation of hypermedia Web APIs (Lanthaler, 2021)

Nevertheless, this thesis covers a significant part of the RDF ecosystem that is relevant today. Thereby, it also indicates that the extent and the resulting complexity of the RDF ecosystem makes it difficult to enter the domain of RDF-based KGs. Therefore, Section 2.1 represents an important contribution as it provides a concise introduction to important aspects of KGs, RDF, and its ecosystem.

One problem that follows from the ecosystem's complexity is that beginners might select a technology without having the capacity to adequately evaluate its potential consequences. Consider Application 3 as an example: While both OWL and SHACL are suitable options for the implementation of QuaRe, as described in [VII], an OWL-based implementation requires more resources and is still significantly slower than a SHACL-based in this particular case. Parts of the community are aware of this problem and attempt to improve the accessibility of the ecosystem. The vocabulary maintained by the schema.org community (Guha et al., 2016) is one example of these efforts as it covers, unlike many other vocabularies, a large semantic range, thus representing an adequate option for many applications.

6.2.4 End-User Interfaces

For experienced software engineers, the triple-based data model of RDF might appear trivial. However, it is important not to be misled into assuming that this also applies to typical end-users. The user study conducted in the context of Application 2 reveals that even LaTeX users with computer science degrees have problems parsing Turtle-formatted RDF data. This is an important observation since Turtle com-

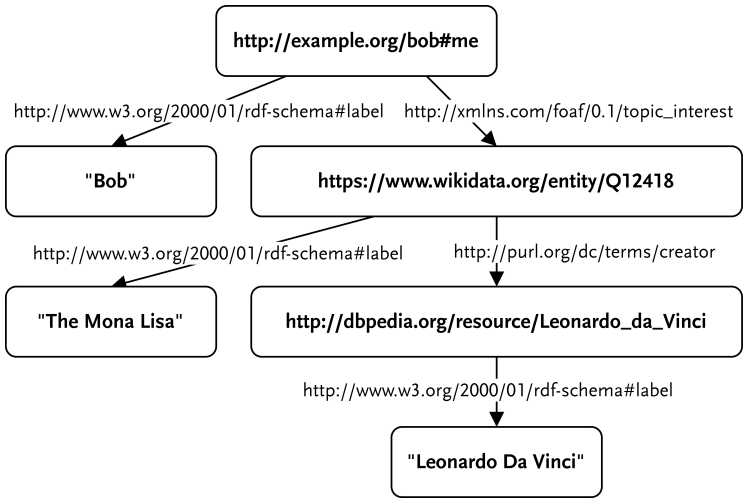


Figure 24: A subgraph of the RDF graph depicted in Figure 6.

```
BASE   <http://example.org/>
PREFIX rdfs: <http://w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX dbpedia: <http://dbpedia.org/resource/>

<bob#me>
    rdfs:label "Bob" ;
    foaf:topic_interest wd:Q12418 .

wd:Q12418
    rdfs:label "Mona Lisa" ;
    dcterms:creator dbpedia:Leonardo_da_Vinci .

dbpedia:Leonardo_da_Vinci
    rdfs:label "Leonardo Da Vinci" .
```

```
BASE   <http://example.org/>
PREFIX rdfs: <http://w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX dbpedia: <http://dbpedia.org/resource/>

dbpedia:Leonardo_da_Vinci
    rdfs:label "Leonardo Da Vinci" .

<bob#me> foaf:topic_interest wd:Q12418 .

wd:Q12418
    dcterms:creator dbpedia:Leonardo_da_Vinci ;
    rdfs:label "Mona Lisa" .

<bob#me> rdfs:label "Bob" .
```

Listing 5: The RDF data from Figure 24 in Turtle format. Left: Comprehensible breadth-first ordering of triples with full use of syntactic sugar. Right: Arbitrary ordering with only partial use of syntactic sugar.

prises various forms of syntactic sugar as it is designed with readability in mind (Schreiber & Raimond, 2014). One contributing factor is that Turtle does not prescribe in which order the triples of an RDF graph appear in the code. As an example, consider Figure 24, which shows a subgraph of the graph depicted in Figure 6. From a human point of view, a serialization format should probably list the triples in an order according to depth-first or breadth-first search. However, Listing 5 shows that the order of triples can, in fact, be arbitrary. Furthermore, even if Turtle offers syntactic sugar, it is not mandatory to use it. In the example on the right side, for example, the triples associated with `<bob#me>` are not combined like on the left side. Depending on the employed libraries, the resulting serialization might therefore differ, potentially impeding the readability of the RDF data. Accordingly, serialization formats should not be leveraged for presenting RDF data to end-users, also not for the presentation of entity relationship explanations in Application 1.

There is also another point to be adressed here: Different technologies in the RDF ecosystem aim to enrich other data with semantic information. One prominent example is RDFa, which facilitates adding semantic information to HTML and other XML-based documents (cf. Section 2.1). While this capability opens up new opportunities for applications, every added piece of information increases the complexity of the code, ultimately impeding its maintainability. The same holds for the RDFtex framework proposed in the context of Application 2, which builds upon the RDFa idea to support the enrichment of LaTeX documents with semantic information. In this regard, the test subjects from the user study criticized the effort associated with the additional usage of the custom RDFtex commands, which only becomes acceptable when the benefits of knowledge graph augmented research become tangible.

6.3 Recommendations

With the goal of answering the final top-level research question 3, the following section investigates several recommendations for the utilization of (RDF-based) KGs. Some of them are addressed to expert users and developers interested in utilizing KG technology for an application, while others are more relevant for the maintainers of KGs.

6.3.1 Implementation of Knowledge Graph Applications

The establishment of a new KG from scratch is a complex endeavour as it includes tasks that require a lot of expertise such as ontology engineering, which is an entire discipline in itself (Suárez-Figueroa et al., 2012). From an organization's point of view, it is therefore worthwile to explore which resources already exist and to what extent they could be reused. In this regard, the first step should be the assessment of available KGs. Publicly available open-domain KGs cover a lot of domains, potentially also the domains of interest. And even if the data does not meet all requirements, it might still be good enough for prototyping envisaged applications. Afterwards, a better understanding of risks and challenges will have emerged and one can still decide to create a new KG. But, there are mixed solutions, as well: As explained in the beginning of Section 6.1, (RDF-based) KGs permit links to external resources by design. Hence, one could create a smaller KG that comprises application-specific data and include links to external KGs where useful. Moreover, KGs such as Wikidata provide another option that eliminates external dependencies: Since the majority of data in Wikidata resides in the public domain through the Creative Commons Licenses[78] CC0 and Attribution-ShareAlike[79], one can download Wikidata dumps[80] and use the data as the foundation of a proprietary KG internally. These mixed solutions can be compared to the common practice of taking an off-the-shelf ML model and fine-tuning it using additional proprietary training data (Zhuang et al., 2021).

In any case, practitioners have to carefully select the vocabularies to be used whenever they create new RDF data manually or when they implement applications that produce RDF data. Following the classic principle *convention over configuration*, they should resort to commonly used vocabularies that already exist. There are various reasons for this:

1. The numerous existing vocabularies can express a lot of semantic information. There are also a number of vocabularies that are already partially redundant. Creating new vocabularies could therefore aggravate this issue, without providing significant benefits.
2. The creation of good vocabularies is no trivial task that requires a lot of expertise and resources, albeit to a lesser extent than the creation of full KGs.
3. The reuse of existing vocabularies facilitates the integration with other KGs.

[78] https://creativecommons.org/share-your-work/cclicenses (visited 2024-07-22)
[79] https://www.wikidata.org/wiki/Wikidata:Licensing (visited 2024-07-22)
[80] https://www.wikidata.org/wiki/Wikidata:Database_download (visited 2024-07-22)

To identify suitable candidate vocabularies, the KnowDive search engine[81] mentioned in Section 2.1 is a recommendable tool. That being said, the implementation of a KG application requires more than appropriate KGs and vocabularies. The technology stacks of the applications examined in this thesis leverage various progamming languages, libraries, architectures, and interfaces. To retain the scope, the following recommendations focus on the RDF ecosystem and omit the discussion of general best practices from software engineering[82].

The complexity of the RDF ecosystem might scare off beginners. But on the contrary, due to number of opportunities the varied technologies from the ecosystem provide, some practictioners might also be inclined to inflate the scope unnecessarily. As an example, consider a simple application whose purpose is to retrieve facts from a domain-specific KG via SPARQL. Since the RDF standards make the addition of other RDF technologies easy, more experienced users might want to introduce technologies for KG completion or import and export mechanism right from the start even though these functionalities are out of scope. As a result, the implementation and maintenance costs would increase without benefits. To avoid this, practictioners should view RDF's extensibility as an option that *can* but not *must* be leveraged. Accordingly, the minimum number of technologies from the RDF ecosystem should be employed to implement an application with its concrete requirements in the beginning. Only when requirements or features are requested that cannot be reasonably realized using the present technology stack, new technology should be added. Although still recommended, adherence to the principle of *design for change* from agile software development is not as important as for other software due to the inherent compatibility of various RDF technologies.

Awareness of the targeted users and their level of expertise is pivotal for the implementation of effective user interfaces with a high usability. Despite the seemingly trivial data model of RDF, only experts should ever be directly confronted with RDF data, disregarding the serialization format. In this regard, Li et al. (2024) present future research directions for the visualization of KGs. In cases where the direct manipulation of RDF data is required, such as in the case of RDFa or RDFtex from Application 2, the provision of syntactic sugar such as the prefix syntax is highly recommended to alleviate the negative impact of additional RDF data on the readability of enriched source code.

6.3.2 Governance Processes

As said before, data quality is an important topic in the context of KGs since many applications heavily depend on the correctness of the data to perform well. Typical data quality issues that could surface in this regard include:

- Redundancy of data including duplicate entities (with overlapping properties)
- Outdated information
- Missing links and inconsistent use of properties

[81] https://lov.linkeddata.es/dataset/lov (visited 2024-05-22)

[82] Of course, the effective and efficient implementation of a KG application requires conducting activities from software engineering, as well. This includes the selection of the software development process, requirements engineering, architectural design, and software testing, among others.

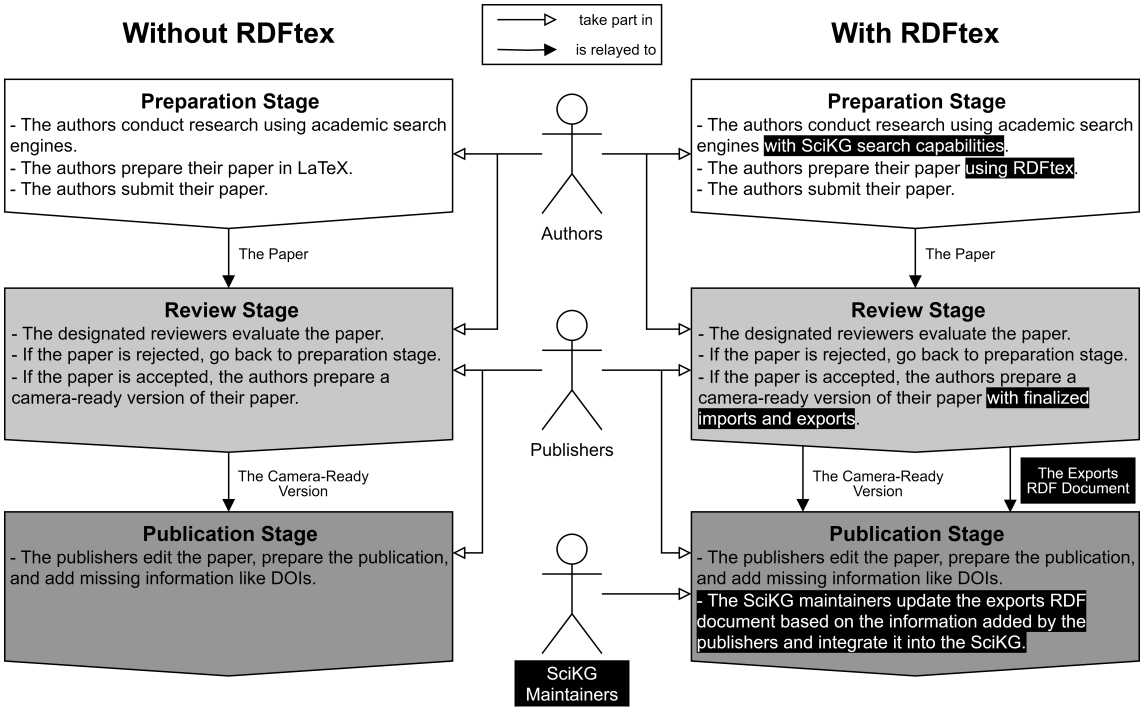


Figure 25: A comparison between a standard three-staged research and publication process on the left with another incorporating RDFtex on the right. The parts with a black background are changes to the process introduced by RDFtex. Figure and caption adopted from [VI].

The establishment of data governance processes is therefore pivotal for the success of KGs. Considering this thesis, Application 2 would be affected by data quality issues the most. Therefore, [VI] includes remarks on data governance in the context of RDFtex and SKGs such as the ORKG. This also comprises the diagram depicted in Figure 25, which proposes a three-staged research and publication process incorporating RDFtex. Notably, the diagram focuses on the obligations of partaking user groups (in this case authors, publishers, and SKG maintainters), which is a central aspect of data governance. In the end, it is in many cases the maintainers' obligation to ensure that data quality requirements are met.

6.3.3 Facilitating Access to Knowledge Graphs

Clearly, maintainers want that their KGs are employed for many applications and that they have a good reputation in the respective user group. To achieve this, the provision of useful data access facilities and interfaces is pivotal. Table 9 gives an overview of the official data access facilities and interfaces Wikidata, YAGO, and the ORKG provide. As shown, each KG provides different options, which is commendable as it allows developers to choose one that fits their application's architecture. The applications examined in this thesis mainly employ SPARQL endpoints. However, leveraging this option is not always feasible: If there was no workaround for the Wikidata Query Service limitations encountered in Application 1, for example, using another option would have been required, potentially even the dumps. But, the full Wikidata dump from 2024-07-18 in the Turtle format compressed as a gzip archive

Table 9: Data access facilities and interfaces of Wikidata, YAGO, and the ORKG. Information retrieved from https://www.wikidata.org/wiki/Wikidata:Data_access, https://yago-knowledge.org/getting-started, and https://orkg.org/data on 2024-07-22.

Knowledge Graph	Data Access Facilities and Interfaces
Wikidata	• Web Interface • Linked Data Interface • Wikidata Query Service (SPARQL Endpoint) • Linked Data Fragments endpoint • Full Size and Partial Dumps • ...
YAGO	• Web Interface • SPARQL Endpoint • Full Size and Partial Dumps
ORKG	• Web Interface • REST API • Python Wrapper for REST API • SPARQL Endpoint • Full Size Dump

is 138.1 GB big[83]. Loading the unpacked archive in an RDF store requires a powerful system with plenty of RAM. Hence, maintainers should take great care to make their interfaces powerful enough for complex queries, thus avoiding discouraging users with hardware limitations.

In contrast to Application 1, many applications only require certain parts of the data large open-domain KGs provide, though. To cover these cases, maintainers should provide partial dumps. Wikidata offically provides a dump without external references. But, with 67.5 GB compressed, this dump is still not easy to handle. Apart from this, there are third-party tools that allow generating filtered dumps[84]. In the case of YAGO 4.5, YAGO's most recent version, there is a full dump and a smaller partial dump[85] available, as well. On the opposite, there is only a full size dump available for the ORKG[86]. Given its current size, this is not problematic, at least not yet. To facilitate prototyping and thus attract users, providers of large KGs should distribute parts of their KGs with certain scopes to alleviate problems arising from the lack powerful enough hardware.

Given that SPARQL endpoints are expensive for the server and that downloadable dumps are expensive for clients, Linked Data Fragments (LDFs) (Verborgh et al., 2014) emerged as a promising alternative. LDFs represent a conceptual framework that aims to distribute the load of query executions between client and server through a sophisticated partitioning strategy. Although Wikidata provides an LDFs endpoint, it is not yet clear whether this concept will receive a wide-spread attention, though. If so,

[83] https://dumps.wikimedia.org/wikidatawiki/entities (visited 2024-07-23)
[84] https://wdumps.toolforge.org (visited 2024-07-23)
[85] https://yago-knowledge.org/data/yago4.5 (visited 2024-07-23)
[86] https://orkg.org/data (visited 2024-07-23)

the provision of partial dumps might not be a strong recommendation anymore as LDFs can cover a majority of the associated scenarios while providing significant advantages. Also worth mentioning in this regard is the CLOCQ toolkit (Christmann et al., 2023), which promises fast and easy access to knowledge bases, in particular RDF-based KGs. To do this, the toolkit takes a what is called *fact*-centric view of the KG of interest. In this context, *facts* are subsets of a KG that consist of a main triple and an optional set of relations that further describe the base triple. Indexing these facts rather than individual triples can lead to a reduction of the search space. Accordingly, the evaluation shows that this approach is able to serve certain queries more efficiently in terms of runtime. Like LDFs, CLOCQ is, however, not yet widely recognized in the community.

6.4 Overview

To summarize the results of the investigation of opportunities, challenges, and recommendations for the utilization of (RDF-based) KGs, Table 10 provides a concise overview. Additionally, it reveals whether the respective opportunities, challenges, and recommendations apply to KGs in general or specifically to RDF-based KGs. As shown, three opportunities, six challenges, and five recommendations have been identified. Five of the in total 14 items apply to KGs in general. To what extent the remaining items can be transferred to KGs based on other data models than RDF is a question left open to future work.

Table 10: An overview of the opportunities, challenges, and recommendations for the utilization of KGs identified based on the investigation of the three presented applications. A white background indicates that the respective statements apply to KGs in general, while statements addressing RDF-based KGs are highlighted in gray.

Opportunities	O1	The rich RDF ecosystem facilitates the management of structured, semi-structured, and unstructured data in distributed environments. The standards and technologies within the ecosystem facilitate the implementation of applications with varied requirements.
	O2	Compared to other means of managing large volumes of heterogeneous data, RDF-based KGs are lean and can thus be added to existing systems retroactively with relative ease.
	O3	The recent interest in KGs, which is fueled by new ML approaches that aim to combine symbolic and subsymbolic technologies, as well as the ongoing support and usage of KGs by big players in the market indicate a promising future for KGs.
Challenges	C1	The number of publicly available and up-to-date open-domain KGs is small. Many KGs, sometimes even of a considerable size, have been deprecated and are not actively maintained or supported anymore. In some cases, not even dumps can be obtained.
	C2	Many applications rely on the correctness of the data within KGs to perform well. Especially applications that require individual facts to be correct are susceptible to data quality issues.
	C3	Beginners might be overwhelmed by the RDF ecosystem as it comprises a large number of technologies and standards. The fact that there are many technologies and standards that have been deprecated or replaced aggravates this issue.
	C4	Certain technologies in the RDF ecosystem can add significant overhead to an application that is difficult for beginners to anticipate.
	C5	Despite the seemingly trivial triple-based data model, even more experienced end-users can have problems when interacting with RDF data.
	C6	Adding semantic information in the form of RDF data to source code (cf. RDFa, RDFtex) can impede both its readability and maintainability.
Recommendations	R1	Existing KGs, vocabularies, and other resources should be reused where possible. If necessary, employed existing resources can be enriched with proprietary data.
	R2	Applications should be implemented using a minimal technology stack, first. The inherent compatibility between different RDF technologies facilitates the iterative extension of existing applications.
	R3	Only expert users should be confronted with RDF data directly. In cases where the interaction with RDF data is inevitable, syntactic sugar should be employed to limit the negative impact of additional RDF data on code and make RDF data more readable, in general.
	R4	Adequate data governance processes should be established to ensure that data quality requirements are met and that obligations of partaking user groups are clear.
	R5	Maintainers should provide multiple options for interacting with RDF-based KGs. At least, a SPARQL endpoint and RDF dumps should be available. In case of large KGs, the provision of partial dumps is recommended to facilitate prototyping.

7 Conclusion

The overarching goal of this doctoral thesis is to contribute to the maturation of RDF-based KGs as technology by deriving opportunities, challenges, and recommendations for their utilization based on an investigation of three novel KG applications. To round of this work, the next paragraphs point out the central contributions, before limitations and future work are addressed in the subsequent section.

The first central contribution of this thesis is the concise introduction to KGs and essential components of the complex RDF ecosystem in Section 2.1. The contained overview provides readers with the fundamental information to dive into the domain of RDF-based KGs and also references useful resources for further guidance. Afterwards, approaching the main topic of the thesis, an analysis of surveys and other meta-level works addressing the utilization of KGs is conducted.

Representing the next set of central contributions, the investigation of the three applications follows in Section 3, Section 4, and Section 5. The presented applications have been selected such that they operate on KGs with varied scopes, i.e., open-domain KGs, domain-specific KGs, and KG fragments. In terms of size, Wikidata, an open-domain KG, which has been employed in Application 1, is by far the largest KG used in this thesis. In contrast, Application 2 and Application 3 both leverage KG fragments, i.e., the smallest KGs employed herein, as representations of research contributions and of software repositories tailored to the respective application. Moreover, each application employs different technologies from the RDF ecosystem and targets different user groups, further widening the capacity of the thesis. As an indication of the communities' interest in the examined applications, the large majority of the research conducted on these applications has been published or is in the process being published as supplementary papers. Both the selection of applications and their investigation can thus be considered successful. In combination with the introductory paper, the supplementary papers constitute this cumulative doctoral thesis.

Based on the insights obtained from the investigation of the applications, several opportunities, challenges, and recommendations have been derived in Section 6 that, in parts, touch upon aspects mentioned in related work and, in other parts, also raise new points. By answering the top-level research questions 1 , 2 , and 3 , this section thus provides the final central contribution of this thesis.

7.1 Limitations and Future Work

As already pointed out in Section 1, the utilization of KGs is a complex topic, only a part of which could be covered in this thesis. Accordingly, one significant but necessary restriction that had to be introduced is the focus on RDF-based KGs. Even if references to KGs based on other data models have been made at various points, it is still necessary to test in practice to what extent the statements with regard to opportunities, challenges, and recommendations can be transferred. Another limitation originates from the selected applications investigated in this thesis. Section 1 explained the requirements ensuring the selection of appropriate applications. Nevertheless, the selection is still biased to a certain degree and

the selected applications only cover a fraction of all aspects of RDF-based KGs relevant regarding opportunities, challenges, and recommendations for their utilization.

Hence, regarding future work, there are two levels to consider. On the one hand, each application investigated in this doctoral thesis left open points that should to be considered for the continuation of the respective line of research. Detailed information on these points can be found in the supplementary papers. For completeness, a selection of important aspects should still be mentioned here:

Application 1 (Dual-Entity Knowledge Panels) Before dual-entity KPs become ready for an end-to-end implementation in a productive web search engine, two large-scale user studies have to be conducted next. So far, the cost function of BiPaSs has been optimized to maximize coverage and follow semantically related entities. How well the current cost function aligns with the expectations and needs of users still has to be evaluated, though. For instance, predicates that express some taxonomical relationship might be more useful for users in comparison to predicates that require domain-specific knowledge. The insights from a user study could be used to further improve the cost function. The second user study aims to obtain additional feedback on the presentation formats for entity relationships. Questions to be answered in this regard relate to the further improvement of the prototypical presentation formats that have already been implemented and to the mix of visual and textual presentation formats.

Application 2 (Import and Export of Research Contributions) To put the bidirectional knowledge exchange established through RDFtex into practice, the most important next step is the adaptation of the tool to a productive SKG. So far, RDFtex has been set up for interaction with a small makeshift SKG as a showcase in order to stimulate community feedback. Since the supplementary papers for this application have been published, papers by other researchers have commended RDFtex's ability to describe research publications semantically during the preparation process and how this integrates with SKG such as the ORKG (Karras et al., 2023). When enough feedback has arrived, RDFtex can be improved accordingly and the adaptation can take place.

Application 3 (Quality Assessment of Software Repositories) While QuaRe's current version is able to validate software repositories against predefined project types consisting of an arbitrary number of quality constraints in a fast and convenient manner, it lacks an end-user friendly interface for customizing and creating said project types and quality constraints. In this regard, the implementation of a form for composing new project types based on already available quality constraints on QuaRe's specification page is straightforward. In contrast, realizing an interface that allows users to create new quality constraints poses significant challenges since the input has to be translated into SHACL shapes, based on which calls to the GitHub API have to be executed. One conceivable approach is to implement a low-code interface that provides enough abstraction such that users do not have to directly interact with the unintuitive SHACL syntax without restricting the variety of producible quality constraints too much. Nevertheless, the further development of the tool heavily depends on user feedback such that adequate templates for API calls can be added.

On the other hand, there is also future work to be done on the level of this doctoral thesis as a whole. These aspects follow from the goal and approach of the thesis. While the three applications presented

herein have been chosen with great care to ensure both novelty and diversity, they cover only a fraction of the extensive RDF ecosystem and the huge potential of KGs in general. As a result, the remarks on opportunities, challenges, and recommendations for the utilization of KGs are limited to the insights these applications offer. To develop a more comprehensive understanding, it is thus essential to explore complementary applications. In this regard, investigating KGs that are not based on RDF is crucial for determining the extent to which the insights gained from applications of RDF-based KGs are transferable, thereby supporting informed decisions regarding the utilization of KG technology. This broader approach will not only fill the existing gaps but also enhance the robustness and applicability of the findings across different KG frameworks. Ultimately, the continued exploration and utilization of KGs hold immense potential for advancing research and driving innovation within organizations.

List of Figures

Figure 1: The number of publications returned in response to the query *knowledge graph* in dblp's record from 2013 to 2023. Data retrieved from https://dblp.uni-trier.de/search?q=knowledge+graph on 2024-07-03. 3
Figure 2: The positioning of KGs in the 2021, 2022, 2023, and 2024 editions of Gartner's Hype Cycle for Artificial Intelligence. Based on Goasduff (2021); Jaffri & Khandabattu (2024); Perri (2023); Wiles (2022). 4
Figure 3: The structure of this doctoral thesis. The arrows illustrate that the insights from the investigation of the selected applications serve as the foundation for deriving opportunities, challenges, and recommendations for the utilization of KGs in this introductory paper. 7
Figure 4: The parts of the RDF ecosystem relevant for this thesis. The depicted model is related to the variations of the semantic web stack (Horrocks et al., 2005) with an additional focus on the relationship between the technologies. 11
Figure 5: A simplified and slightly adapted version of the introductory RDF graph shown in the RDF documentation (Schreiber & Raimond, 2014). 12
Figure 6: A more accurate visualization of the RDF graph from Figure 5 with additional triples encoding the entity labels explicitly. 13
Figure 7: The results of the hyperparameter optimization. The y-axis denotes the average number of entities visited by the configuration examined in the respective iteration, i.e., the *objectiveValue*. The *minValue* equals the minimal *objectiveValue* encountered up to the respective iteration. Figure adopted from [II]. 34
Figure 8: The user interface of the testbed after clicking the *Generate Knowledge Panel* button. In this screenshot, the lengthy Turtle document encoding the path is not shown completely. Figure and caption adopted from [III]. 36
Figure 9: The four currently implemented prototypical presentation formats. Figures and captions adopted from [III]. 37
Figure 10: An example path visualized using the presentation format based on node-link diagrams as proposed in Brand (2023). The information button was clicked, revealing additional information on the meaning of the displayed colors. 39
Figure 11: The number of publications added to dblp's record *per year* from 2008 to 2023. Data retrieved from https://dblp.org/statistics/publicationsperyear.html on 2024-07-03. 41
Figure 12: A flowchart of RDFtex's preprocessing workflow. For conciseness, the injection of custom LaTeX environments, which are necessary for importing some of the supported contribution types, was omitted. Figure and caption adopted from [VI]. 45
Figure 13: A simplified illustration of QuaRe's validation process with focus on the generation of representations of the repositories of interest, which are called data graphs in the SHACL context. Figure adopted from [VIII]. 50
Figure 14: Left: A part of a shapes graph defining the quality criterion that a repository containing a finished research project must possess exactly one branch. Right: A part of a data graph derived from

a GitHub repository. Figures adopted from [VII]. 50
Figure 15: The results of the initial runtime evaluation of QuaRe. Apart from the employed approach, the x-axis mentions the examined project types: T_F is a project type comprising eleven quality criteria and T_I another one comprising four quality criteria. Figure adopted from [VII]. 51
Figure 16: A modified version of the graphs from Figure 14. Left: A part of a shapes graph defining the quality criterion that a repository containing a finished research project must possess exactly one branch. Right: A part of a data graph derived from a GitHub repository. 52
Figure 17: The fragment of the SHACL shapes graph representing the FAIRSoftware project type; other project types are omitted here. The project type shape for the FAIRSoftware project type is depicted in blue, the corresponding node and property shapes are light yellow and turquoise. *sh* refers to the SHACL namespace. Figure and caption adopted from [VIII]. 53
Figure 18: An abstract visualization of the ontology underlying the repository representations (data graphs). IRIs are depicted in blue, literals in green. The cardinality is given in brackets. *sd* refers to the Software Description Ontology namespace, and *props* to the namespace for additional custom properties. Figure and caption adopted from [VIII]. 53
Figure 19: A screenshot of the validation page: In response to a click on the submit button, a repository has been validated against the new project type *FAIRSoftware*. Afterwards, the button labeled *View* was clicked, revealing the raw and verbalized explanations. Figure and caption adopted from [VIII]. 54
Figure 20: A screenshot of QuaRe's specification page showing the specification of two quality criteria. Figure and caption adopted from [VIII]. 56
Figure 21: A simplified illustration of how KGs and data lakes integrate data. The example use case is to ingest a document from an e-commerce platform to extract prices and pictures of products. The shown KG uses properties from the schema.org vocabulary. 62
Figure 22: A simplified illustration of how the RAG models retrieve external knowledge to enhance the LLM generation. Figure and caption adapted from Pan et al. (2023). 64
Figure 23: A selection of the information offered by Wikidata on the entity *Paris* (Q167646). The prefix rdfs is short for http://www.w3.org/2000/01/rdf-schema#, schema for http://schema.org/ and wdt for http://www.wikidata.org/prop/direct/. Data retrieved from https://www.wikidata.org/wiki/Q167646 on 2024-07-18. 68
Figure 24: A subgraph of the RDF graph depicted in Figure 6. 69
Figure 25: A comparison between a standard three-staged research and publication process on the left with another incorporating RDFtex on the right. The parts with a black background are changes to the process introduced by RDFtex. Figure and caption adopted from [VI]. 73

List of Tables

Table 1: An overview of the three selected applications denoting the title of the applications as well as the scope(s) of and task(s) performed on the respectively employed KGs. The shown scope terminology will be further explained in Section 2. 6
Table 2: An overview of the three examined applications and the corresponding GitHub repositories that comprise the implemented software and the associated research data. 8
Table 3: The result returned by the Wikidata Query Service in response to the query from Listing 3. Data retrieved from https://w.wiki/8p2o on 2024-01-11. 20
Table 4: A subset of the information displayed in the knowledge panel variants of Google, Bing, and Startpage when the two queries *European Union* and *Alan Turing* are issued individually. Information from third-party sources like weather services has been left out. The search engines were set to English and the searches were conducted on 2023-05-26. Table and caption adapted from [II]. 27
Table 5: Comparison of semantic distances using five examples with two Wikidata entities each. d_{labels} and $d_{labels+descs}$ denote whether the presented semantic distances were calculated using the cosine distance between SBERT vector representations of the entity labels alone or of the concatenated entity labels and entity descriptions. Table and caption adapted from [II]. 33
Table 6: An overview of RDFtex's four custom LaTeX commands, their purpose and the required parameters. Table and caption adapted from Martin & Henrich (2023b). 44
Table 7: An example of how a best practice that is derived from an abstract FAIR principle is translated into a compound rule that can implemented in SHACL. Based on [VIII]. 58
Table 8: Size and order statistics of the KGs Wikidata, DBpedia, Freebase, and YAGO. Numbers adopted from Färber et al. (2018). 66
Table 9: Data access facilities and interfaces of Wikidata, YAGO, and the ORKG. Information retrieved from https://www.wikidata.org/wiki/Wikidata:Data_access, https://yago-knowledge.org/getting-started, and https://orkg.org/data on 2024-07-22. 74
Table 10: An overview of the opportunities, challenges, and recommendations for the utilization of KGs identified based on the investigation of the three presented applications. A white background indicates that the respective statements apply to KGs in general, while statements addressing RDF-based KGs are highlighted in gray. 75

List of Acronyms

AI:	Aritificial Intelligence
CCS:	ACM Computing Classification System
CRUD:	Create, Read, Update, Delete
DCMI:	Dublin Core Metadata Initiative
DOI:	Digital Object Identifier
FOAF:	Friend of a Friend
IRI:	Internationalized Resource Identifier
KG:	Knowledge Graph
KMS:	Knowledge Management System
KP:	Knowledge Panel
LLM:	Large Language Model
LDF:	Linked Data Fragment
ML:	Machine Learning
ORKG:	Open Research Knowledge Graph
OWL:	Web Ontology Language
RAG:	Retrieval-Augmented Generation
RDF:	Resource Description Framework
RDFa:	RDF in Attributes
RDFS:	RDF Schema
SKG:	Scientific Knowledge Graph
SKOS:	Simple Knowledge Organization System
Turtle:	Terse RDF Triple Language
SHACL:	Shapes Constraint Language
SPARQL:	SPARQL Protocol And RDF Query Language
URI:	Uniform Resource Identifier
URL:	Uniform Resource Locator
UUID:	Universally Unique IDentifier

Bibliography

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Q.*, *25*(1), 107–136. http://misq.org/review-knowledge-management-and-knowledge-management-systems-conceptual-foundations-and-research-issues.html

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. G. (2007). DBpedia: A Nucleus for a Web of Open Data. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, & P. Cudré-Mauroux (Eds.), *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007* (Vol. 4825, pp. 722–735). Springer. https://doi.org/10.1007/978-3-540-76298-0_52

Auer, S., Kovtun, V., Prinz, M., Kasprzik, A., Stocker, M., & Vidal, M.-E. (2018). Towards a Knowledge Graph for Science. *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018*, 1–6. https://doi.org/10.1145/3227609.3227689

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press. https://doi.org/10.1017/CBO9780511711787

Bast, H., Bäurle, F., Buchhold, B., & Haußmann, E. (2014). Easy access to the freebase dataset. In C.-W. Chung, A. Z. Broder, K. Shim, & T. Suel (Eds.), *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume* (pp. 95–98). ACM. https://doi.org/10.1145/2567948.2577016

Beckett, D., Berners-Lee, T., Prud'hommeaux, E., & Carothers, G. (2014). RDF 1.1 Turtle. *W3c Recommendation*. https://www.w3.org/TR/2014/REC-turtle-20140225

Belotti, M., Bozic, N., Pujolle, G., & Secci, S. (2019). A Vademecum on Blockchain Technologies: When, Which, and How. *IEEE Commun. Surv. Tutorials*, *21*(4), 3796–3838. https://doi.org/10.1109/COMST.2019.2928178

Bing. (2023). *Bing Preview Release Notes: AI-powered Knowledge Cards and Stories*. https://blogs.bing.com/search/march_2023/Bing-Preview-Release-Notes-AI-powered-Knowledge-Cards-and-Stories

Bizer, C., Heath, T., & Berners-Lee, T. (2023). Linked Data - The Story So Far. In O. Seneviratne & J. A. Hendler (Eds.), *Linking the World's Information - Essays on Tim Berners-Lee's Invention of the World Wide Web* (Vol. 52, pp. 115–143). ACM. https://doi.org/10.1145/3591366.3591378

Bless, C., Baimuratov, I., & Karras, O. (2023). SciKGTeX - A LATEX Package to Semantically Annotate Contributions in Scientific Publications. *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2023, Santa Fe, NM, USA, June 26-30, 2023*, 155–164. https://doi.org/10.1109/JCDL57899.2023.00030

Blumberg, R., & Atre, S. (2003). The problem with unstructured data. *Dm Review*, *13*(42–49), 62–63.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bollobás, B. (2002). *Modern Graph Theory* (Vol. 184). Springer. https://doi.org/10.1007/978-1-4612-0619-4

Bonatti, P. A., Decker, S., Polleres, A., & Presutti, V. (2018). Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371). *Dagstuhl Reports*, *8*(9), 29–111. https://doi.org/10.4230/DagRep.8.9.29

Brack, A., Hoppe, A., Stocker, M., Auer, S., & Ewerth, R. (2020). Requirements Analysis for an Open Research Knowledge Graph. *Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25-27, 2020, Proceedings*, *12246*, 3–18. https://doi.org/10.1007/978-3-030-54956-5_1

Brand, S. M. (2023). *Visualization of Relationships Between Entities in Knowledge Graphs to Enhance Knowledge Panels*. Supervisor: Martin, Leon; Reviewer: Henrich, Andreas. University of Bamberg.

Brickley, D., & Miller, L. (2014). FOAF Vocabulary Specification 0.99. *Namespace Document*. http://xmlns.com/foaf/spec/20140114.html

Brickley, D., Guha, R. V., & McBride, B. (2014). RDF Schema 1.1. *W3c Recommendation*. http://www.w3.org/TR/2014/REC-rdf-schema-20140225

Cantallops, M. M., Sánchez-Alonso, S., & García-Barriocanal, E. (2019). A systematic literature review on Wikidata. *Data Technol. Appl.*, *53*(3), 250–268. https://doi.org/10.1108/DTA-12-2018-0110

Cao, N. D., Izacard, G., Riedel, S., & Petroni, F. (2021). Autoregressive Entity Retrieval. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. https://openreview.net/forum?id=5k8F6UU39V

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., & Duan, Z. (2020). Knowledge Graph Completion: A Review. *IEEE Access*, *8*, 192435–192456. https://doi.org/10.1109/ACCESS.2020.3030076

Chessell, M., Scheepers, F., Nguyen, N., Kessel, R. van, & Starre, R. van der. (2014). Governing and managing big data for analytics and decision makers. *IBM Redguides for Business Leaders*, *252*. https://www.redbooks.ibm.com/redpapers/pdfs/redp5120.pdf

Choo, C. W., Detlor, B., & Turnbull, D. (2000). Information Seeking on the Web: An Integrated Model of Browsing and Searching. *First Monday*, *5*(2). https://doi.org/10.5210/fm.v5i2.729

Christmann, P., Roy, R. S., & Weikum, G. (2023). CLOCQ: A Toolkit for Fast and Easy Access to Knowledge Bases. In B. König-Ries, S. Scherzinger, W. Lehner, & G. Vossen (Eds.), *Datenbanksysteme für Business, Technologie und Web (BTW 2023), 20. Fachtagung des GI-Fachbereichs ,,Datenbanken und Informationssysteme" (DBIS), 06.-10, März 2023, Dresden, Germany, Proceedings* (pp. 579–591). Gesellschaft für Informatik e.V. https://doi.org/10.18420/BTW2023-28

Connolly, D., Harmelen, F. van, Horrocks, I., McGuiness, D. L., Patel-Schneider, P. F., & Stein, L. A. (2001). DAML+OIL (March 2001) Reference Description. *W3c Note*. https://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, 3rd Edition*. MIT Press. http://mitpress.mit.edu/books/introduction-algorithms

Cyganiak, R., Hyland-Wood, D., & Lanthaler, M. (2014). RDF 1.1 Concepts and Abstract Syntax. *W3c Recommendation*. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225

De Jong, T., & Ferguson-Hessler, M. G. (1996). Types and Qualities of Knowledge. *Educational Psychologist, 31*(2), 105–113. https://doi.org/10.1207/s15326985ep3102_2

Dedehayir, O., & Steinert, M. (2016). The hype cycle model: A review and future directions. *Technological Forecasting and Social Change, 108*, 28–41. https://doi.org/10.1016/j.techfore.2016.04.005

Deutsche Forschungsgemeinschaft. (2022). *Guidelines for Safeguarding Good Research Practice. Code of Conduct*. Zenodo. https://doi.org/10.5281/ZENODO.6472827

Dietz, L., Kotov, A., & Meij, E. (2018). Utilizing Knowledge Graphs for Text-Centric Information Retrieval. In K. Collins-Thompson, Q. Mei, B. D. Davison, Y. Liu, & E. Yilmaz (Eds.), *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018* (pp. 1387–1390). ACM. https://doi.org/10.1145/3209978.3210187

Dublin Core Metadata Initiative. (2023). *DCMI Metadata expressed in RDF Schema Language*. https://www.dublincore.org/schemas/rdfs

Dumontier, M. (2022). A formalization of one of the main claims of "The FAIR Guiding Principles for scientific data management and stewardship" by Wilkinson et al. 20161. *Data Sci., 5*(1), 53–56. https://doi.org/10.3233/ds-210047

Dürst, M. J., & Suignard, M. (2005). Internationalized Resource Identifiers (IRIs). *RFC, 3987*, 1–46. https://doi.org/10.17487/RFC3987

Ehrlinger, L., & Wöß, W. (2016). Towards a Definition of Knowledge Graphs. In M. Martin, M. Cuquet, & E. Folmer (Eds.), *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), September 12-15, 2016* (Vol. 1695). CEUR-WS.org. http://ceur-ws.org/Vol-1695/paper4.pdf

Ernst, P., Meng, C., Siu, A., & Weikum, G. (2014). KnowLife: A knowledge graph for health and life sciences. In I. F. Cruz, E. Ferrari, Y. Tao, E. Bertino, & G. Trajcevski (Eds.), *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014* (pp. 1254–1257). IEEE Computer Society. https://doi.org/10.1109/ICDE.2014.6816754

Fernández, J. D., Martínez-Prieto, M. A., Gutierrez, C., Polleres, A., & Arias, M. (2013). Binary RDF representation for publication and exchange (HDT). *J. Web Semant., 19*, 22–41. https://doi.org/10.1016/j.websem.2013.01.002

Färber, M., Bartscherer, F., Menne, C., & Rettinger, A. (2018). Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web, 9*(1), 77–129. https://doi.org/10.3233/SW-170275

Goasduff, L. (2021). *The 4 Trends That Prevail on the Gartner Hype Cycle for AI, 2021*. https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021

Guarino, N., Oberle, D., & Staab, S. (2009). What Is an Ontology?. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 1–17). Springer. https://doi.org/10.1007/978-3-540-92673-3_0

Guha, R. V., Brickley, D., & Macbeth, S. (2016). Schema.org: evolution of structured data on the web. *Commun. ACM, 59*(2), 44–51. https://doi.org/10.1145/2844544

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern., 4*(2), 100–107. https://doi.org/10.1109/TSSC.1968.300136

Hasselbring, W., Carr, L., Hettrick, S., Packer, H. S., & Tiropanis, T. (2020). From FAIR research data toward FAIR and open research software. *It Inf. Technol., 62*(1), 39–47. https://doi.org/10.1515/itit-2019-0040

Hayes, P. J., & Patel-Schneider, P. F. (2014). RDF 1.1 Semantics. *W3c Recommendation*. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225

Herman, I., Adida, B., Sporny, M., & Birbeck, M. (2015). RDFa 1.1 Primer - Third Edition. *W3c Working Group Note*. http://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317

Herman, I., Melançon, G., & Marshall, M. S. (2000). Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Trans. Vis. Comput. Graph., 6*(1), 24–43. https://doi.org/10.1109/2945.841119

Hitzler, P., & Sarker, M. K. (Eds.). (2021). *Neuro-Symbolic Artificial Intelligence: The State of the Art* (Vol. 342). IOS Press. https://doi.org/10.3233/FAIA342

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., & Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition). *W3c Recommendation*. http://www.w3.org/TR/2012/REC-owl2-primer-20121211

Hogan, A. (2020). *The Web of Data*. Springer. https://doi.org/10.1007/978-3-030-51580-5

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. de, Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J. F., Staab, S., & Zimmermann, A. (2021). Knowledge Graphs. *ACM Comput. Surv., 54*(4), 1–37. https://doi.org/10.1145/3447772

Horrocks, I., Parsia, B., Patel-Schneider, P. F., & Hendler, J. A. (2005). Semantic Web Architecture: Stack or Two Towers?. In F. Fages & S. Soliman (Eds.), *Principles and Practice of Semantic Web Reasoning, Third International Workshop, PPSWR 2005, Dagstuhl Castle, Germany, September 11-16, 2005, Proceedings* (Vol. 3703, pp. 37–41). Springer. https://doi.org/10.1007/11552222_4

Hummel, T., Martin, L., & Henrich, A. (2024). Assessing the FAIRness of Software Repositories Using RDF and SHACL. In A. A. Salatino, M. Alam, F. Ongenae, S. Vahdati, A. L. Gentile, T. Pellegrini, & S. Jiang (Eds.), *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands* (Vol. 60, pp. 160–175). IOS Press. https://doi.org/10.3233/SSW240014

Hutson, M. (2018). *Artificial intelligence faces reproducibility crisis*. American Association for the Advancement of Science. https://doi.org/10.1126/science.359.6377.725

Iglesias-Molina, A., & Garijo, D. (2023). Towards Assessing FAIR Research Software Best Practices in an Organization Using RDF-star. In N. Keshan, S. Neumaier, A. L. Gentile, & S. Vahdati (Eds.), *Proceedings of the Posters and Demo Track of the 19th International Conference on Semantic Systems co-located with 19th International Conference on Semantic Systems (SEMANTiCS 2023), Leipzing, Germany, September 20 to 22, 2023* (Vol. 3526). CEUR-WS.org. https://ceur-ws.org/Vol-3526/paper-09.pdf

Jaffri, A., & Khandabattu, H. (2024). *What's New in Artificial Intelligence from the 2023 Gartner Hype Cycle*. https://www.gartner.com/en/documents/5505695

Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., Stocker, M., & Auer, S. (2019). Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, 243–246. https://doi.org/10.1145/3360901.3364435

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Networks Learn. Syst.*, *33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kagdi, H. H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. *J. Softw. Maintenance Res. Pract.*, *19*(2), 77–131. https://doi.org/10.1002/smr.344

Karras, O., Wernlein, F., Klünder, J., & Auer, S. (2023). Divide and Conquer the EmpiRE: A Community-Maintainable Knowledge Graph of Empirical Research in Requirements Engineering. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2023, New Orleans, LA, USA, October 26-27, 2023*, 1–12. https://doi.org/10.1109/ESEM56168.2023.10304795

Kasneci, G., Elbassuoni, S., & Weikum, G. (2009). MING: mining informative entity relationship subgraphs. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, & J. Lin (Eds.), *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009* (pp. 1653–1656). ACM. https://doi.org/10.1145/1645953.1646196

Knublauch, H., & Kontokostas, D. (2017). Shapes Constraint Language (SHACL). *W3c Recommendation*. https://www.w3.org/TR/shacl

Kuhn, T., Meroño-Peñuela, A., Malic, A., Poelen, J. H., Hurlbert, A. H., Ortiz, E. C., Furlong, L. I., Queralt-Rosinach, N., Chichester, C., Banda, J. M., Willighagen, E. L., Ehrhart, F., Evelo, C. T. A., Malas, T. B., & Dumontier, M. (2018). Nanopublications: A Growing Resource of Provenance-Centric Scientific Linked Data. *14th IEEE International Conference on E-Science, E-Science 2018, Amsterdam, The Netherlands, October 29 - November 1, 2018*, 83–92. https://doi.org/10.1109/eScience.2018.00024

Lanthaler, M. (2021). Hydra Core Vocabulary. *W3c Draft*. https://www.hydra-cg.com/spec/latest/core

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, & H.-T. Lin (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Pro-*

cessing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

Li, H. X., Appleby, G., Brumar, C. D., Chang, R., & Suh, A. (2024). Knowledge Graphs in Practice: Characterizing their Users, Challenges, and Visualization Opportunities. *IEEE Trans. Vis. Comput. Graph.*, *30*(1), 584–594. https://doi.org/10.1109/TVCG.2023.3326904

Luan, Y., He, L., Ostendorf, M., & Hajishirzi, H. (2018). Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 3219–3232. https://doi.org/10.18653/v1/d18-1360

Malhotra, A., Peterson, D., Gao, S., Biron, P. V., Sperberg-McQueen, M., & Thompson, H. (2012). W3C XML Schema Definition Language (XSD) 1. 1 Part 2: Datatypes. *W3c Recommendation*. http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405

Martin, L. (2023). BiPaSs: Further Investigation of Fast Pathfinding in Wikidata. In M. Acosta, S. Peroni, S. Vahdati, A. L. Gentile, T. Pellegrini, & J.-C. Kalo (Eds.), *Knowledge Graphs: Semantics, Machine Learning, and Languages - Proceedings of the 19th International Conference on Semantic Systems, 20-22 September 2023, Leipzig, Germany* (Vol. 56, pp. 110–126). IOS Press. https://doi.org/10.3233/SSW230009

Martin, L., & Henrich, A. (2022a). RDFtex: Knowledge Exchange Between LaTeX-Based Research Publications and Scientific Knowledge Graphs. In G. Silvello, Ó. Corcho, P. Manghi, G. M. D. Nunzio, K. Golub, N. Ferro, & A. Poggi (Eds.), *Linking Theory and Practice of Digital Libraries - 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20-23, 2022, Proceedings* (Vol. 13541, pp. 26–38). Springer. https://doi.org/10.1007/978-3-031-16802-4_3

Martin, L., & Henrich, A. (2022b). Specification and Validation of Quality Criteria for Git Repositories using RDF and SHACL. In P. Reuss, V. Eisenstadt, J. M. Schönborn, & J. Schäfer (Eds.), *Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, Hildesheim (Germany), Oktober 5-7th, 2022* (Vol. 3341, pp. 124–135). CEUR-WS.org. https://ceur-ws.org/Vol-3341/WM-LWDA_2022_CRC_1149.pdf

Martin, L., & Henrich, A. (2023a). A Testbed for Dual-Entity Knowledge Panels. In M. Leyer & J. Wichmann (Eds.), *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings, Marburg, Germany, October 9-11, 2023* (Vol. 3630, pp. 231–238). CEUR-WS.org. https://ceur-ws.org/Vol-3630/LWDA2023-paper21.pdf

Martin, L., & Henrich, A. (2023b). RDFtex in-depth: knowledge exchange between LATEX-based research publications and Scientific Knowledge Graphs. *IJDL: International Journal on Digital Libraries*, 1–19. https://doi.org/10.1007/s00799-023-00370-5

Martin, L., Boockmann, J. H., & Henrich, A. (2020). Fast Pathfinding in Knowledge Graphs Using Word Embeddings. In U. Schmid, F. Klügl, & D. Wolter (Eds.), *KI 2020: Advances in Artificial Intelligence - 43rd German Conference on AI, Bamberg, Germany, September 21-25, 2020, Proceedings* (Vol. 12325, pp. 305–312). Springer. https://doi.org/10.1007/978-3-030-58285-2_27

Martin, L., Jegan, R., & Henrich, A. (2021). On the Form of Research Publications for Use in Scientific Knowledge Graphs. *Wissensorganisation 2021: 16. Tagung Der Deutschen Sektion Der Internationalen Gesellschaft Für Wissensorganisation (ISKO) (Wissorg'21)*, accepted for publication, preprint available. https://easychair.org/publications/preprint/SDQs

Mathis, C. (2017). Data Lakes. *Datenbank-Spektrum*, *17*(3), 289–293. https://doi.org/10.1007/s13222-017-0272-7

Miles, A., & Bechhofer, S. (2009). SKOS Simple Knowledge Organization System Reference. *W3c Recommendation*. http://www.w3.org/TR/2009/REC-skos-reference-20090818

Miles, A., Matthews, B., Wilson, M. D., & Brickley, D. (2005). SKOS Core: Simple knowledge organisation for the Web. In T. Baker & E. M. M. Rodríguez (Eds.), *Vocabularies in Practice: Proceedings of the 2005 International Conference on Dublin Core and Metadata Applications, DC 2005, Madrid, Spain, September 12-15, 2005* (pp. 3–10). Dublin Core Metadata Initiative. http://dcpapers.dublincore.org/pubs/article/view/798

Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., & Taylor, J. (2019). Industry-scale Knowledge Graphs: Lessons and Challenges: Five diverse technology companies show how it's done. *Queue*, *17*(2), 48–75. https://doi.org/10.1145/3329781.3332266

Oelen, A., Jaradeh, M. Y., Farfar, K. E., Stocker, M., & Auer, S. (2019). Comparing Research Contributions in a Scholarly Knowledge Graph. *Proceedings of the Third International Workshop on Capturing Scientific Knowledge Co-Located with the 10th International Conference on Knowledge Capture (K-CAP 2019), Marina Del Rey, California , November 19th, 2019, 2526*, 21–26. https://doi.org/10.15488/9388

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2023). Unifying Large Language Models and Knowledge Graphs: A Roadmap. *Corr*. https://doi.org/10.48550/arXiv.2306.08302

Perri, L. (2023). *What's New in Artificial Intelligence from the 2023 Gartner Hype Cycle*. https://www.gartner.com/en/articles/what-s-new-in-artificial-intelligence-from-the-2023-gartner-hype-cycle

Pflaum, L. D. (2022). *Potenziale und Herausforderungen von Wissensgraphen in GitLab im universitären Umfeld*. Supervisor: Martin, Leon; Reviewer: Henrich, Andreas. University of Bamberg.

Pokorný, J. (2015). Graph Databases: Their Power and Limitations. In K. Saeed & W. Homenda (Eds.), *Computer Information Systems and Industrial Management - 14th IFIP TC 8 International Conference, CISIM 2015, Warsaw, Poland, September 24-26, 2015. Proceedings* (Vol. 9339, pp. 58–69). Springer. https://doi.org/10.1007/978-3-319-24369-6_5

Provost, F. J., & Fawcett, T. (2013). Data Science and its Relationship to Big Data and Data-Driven Decision Making. *Big Data*, *1*(1), 51–59. https://doi.org/10.1089/big.2013.1508

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *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* (pp. 3980–3990). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

Reinanda, R., Meij, E., & Rijke, M. de. (2020). Knowledge Graphs: An Information Retrieval Perspective. *Found. Trends Inf. Retr., 14*(4), 289–444. https://doi.org/10.1561/1500000063

Richens, R. H. (1956). Preprogramming for mechanical translation. *Mech. Transl. Comput. Linguistics, 3*(1), 20–25. https://www.mt-archive.net/50/MT-1956-Richens.pdf

Rusher, J. (2001). Triple Store. *W3c Position*. https://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/rusher.html

Saleem, M., Szárnyas, G., Conrads, F., Bukhari, S. A. C., Mehmood, Q., & Ngomo, A.-C. N. (2019). How Representative Is a SPARQL Benchmark? An Analysis of RDF Triplestore Benchmarks. In L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, & L. Zia (Eds.), *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019* (pp. 1623–1633). ACM. https://doi.org/10.1145/3308558.3313556

Schneider, E. W. (1973). *Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis*. https://files.eric.ed.gov/fulltext/ED088424.pdf

Schneider, M., Carroll, J., Herman, I., & Patel-Schneider, P. F. (2012). OWL 2 Web Ontology Language RDF-Based Semantics (Second Edition). *W3c Recommendation*. https://www.w3.org/TR/2012/REC-owl2-rdf-based-semantics-20121211

Schreiber, G., & Raimond, Y. (2014). RDF 1.1 Primer. *W3c Working Group Note*. http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624

Seaborne, A. (2004). RDQL - A Query Language for RDF. *W3c Member Submission*. https://www.w3.org/submissions/2004/SUBM-RDQL-20040109

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intell. Syst., 21*(3), 96–101. https://doi.org/10.1109/MIS.2006.62

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official Google Blog, 5*(16), 3–4. https://blog.google/products/search/introducing-knowledge-graph-things-not

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., & Gangemi, A. (2012). Introduction: Ontology Engineering in a Networked World. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 1–6). Springer. https://doi.org/10.1007/978-3-642-24794-1_1

Tanon, T. P., Weikum, G., & Suchanek, F. M. (2020). YAGO 4: A Reason-able Knowledge Base. In A. Harth, S. Kirrane, A.-C. N. Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, & M. Cochez (Eds.), *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion,Crete, Greece, May 31-June 4, 2020, Proceedings* (Vol. 12123, pp. 583–596). Springer. https://doi.org/10.1007/978-3-030-49461-2_34

Thaller, T. S. (2021). *Konzeption und prototypische Umsetzung einer Anwendung zur Darstellung und Navigation großer RDF-Graphen*. Supervisor: Martin, Leon; Reviewer: Henrich, Andreas. University of Bamberg.

Truica, C.-O., Radulescu, F., Boicea, A., & Bucur, I. (2015). Performance Evaluation for CRUD Operations in Asynchronously Replicated Document Oriented Database. *20th International Conference on*

Control Systems and Computer Science, CSCS 2015, Bucharest, Romania, May 27-29, 2015, 191–196. https://doi.org/10.1109/CSCS.2015.32

Udaly, J. F. (2023). *Verbalization of Heterogeneous Paths in Wikidata using Large Language Models*. Supervisor: Martin, Leon; Reviewer: Henrich, Andreas. University of Bamberg.

Uschold, M., & Gruninger, M. (2009). Ontologies: principles, methods and applications. *Knowl. Eng. Rev.*, *11*(2), 93–136. https://doi.org/10.1017/S0269888900007797

Vandenbussche, P.-Y., Atemezing, G., Poveda-Villalón, M., & Vatant, B. (2017). Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semantic Web*, *8*(3), 437–452. https://doi.org/10.3233/SW-160213

Verborgh, R., Sande, M. V., Colpaert, P., Coppens, S., Mannens, E., & Walle, R. V. de. (2014). Web-Scale Querying through Linked Data Fragments. In C. Bizer, T. Heath, S. Auer, & T. Berners-Lee (Eds.), *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014* (Vol. 1184). CEUR-WS.org. https://ceur-ws.org/Vol-1184/ldow2014_paper_04.pdf

Vrandecic, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Commun. ACM*, *57*(10), 78–85. https://doi.org/10.1145/2629489

W3C SPARQL Working Group. (2013). SPARQL 1.1 Overview. *W3c Recommendation*. http://www.w3.org/TR/2013/REC-sparql11-overview-20130321

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.*, *29*(12), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., & Yu, P. S. (2019). Heterogeneous Graph Attention Network. In L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, & L. Zia (Eds.), *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019* (pp. 2022–2032). ACM. https://doi.org/10.1145/3308558.3313562

White, K. (2019). Publications Output: US Trends and International Comparisons. Science & Engineering Indicators 2020. NSB-2020-6. *National Science Foundation*. https://ncses.nsf.gov/pubs/nsb20206

Wiles, J. (2022). *What's New in Artificial Intelligence from the 2022 Gartner Hype Cycle*. https://www.gartner.com/en/articles/what-s-new-in-artificial-intelligence-from-the-2022-gartner-hype-cycle

Yu, C., Liu, X., Maia, J., Li, Y., Cao, T., Gao, Y., Song, Y., Goutam, R., Zhang, H., Yin, B., & Li, Z. (2024). COSMO: A Large-Scale E-commerce Common Sense Knowledge Generation and Serving System at Amazon. In P. Barceló, N. S. Pi, A. Meliou, & S. Sudarshan (Eds.), *Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024, Santiago AA, Chile, June 9-15, 2024* (pp. 148–160). ACM. https://doi.org/10.1145/3626246.3653398

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., ... Wen, J.-R. (2023). A Survey of Large Language Models. *Corr*. https://doi.org/10.48550/arXiv.2303.18223

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proc. IEEE, 109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Zou, X. (2020). A survey on application of knowledge graph. *Journal of Physics: Conference Series, 1487*(1), 12016–12017. https://doi.org/10.1088/1742-6596/1487/1/012016

Paper I

Fast Pathfinding in Knowledge Graphs Using Word Embeddings

Martin, L., Boockmann, J. H., & Henrich, A. (2020). Fast Pathfinding in Knowledge Graphs Using Word Embeddings. In U. Schmid, F. Klügl, & D. Wolter (Eds.), *KI 2020: Advances in Artificial Intelligence - 43rd German Conference on AI, Bamberg, Germany, September 21-25, 2020, Proceedings* (Vol. 12325, pp. 305–312). Springer. https://doi.org/10.1007/978-3-030-58285-2_27

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (70%)
  - Review of related work (80%)
  - Design of the bidirectional A* pathfinding algorithm including the cost function (80%)
  - Implementation of the pathfinding algorithm including the cost function (60%)
  - Evaluation of the pathfinding algorithm based on a manually created set of dual-entity queries (70%)
  - Discussion of limitations and future work (70%)

Paper II

BiPaSs:
Further Investigation of Fast Pathfinding in Wikidata

Martin, L. (2023). BiPaSs: Further Investigation of Fast Pathfinding in Wikidata. In M. Acosta, S. Peroni, S. Vahdati, A. L. Gentile, T. Pellegrini, & J.-C. Kalo (Eds.), *Knowledge Graphs: Semantics, Machine Learning, and Languages - Proceedings of the 19th International Conference on Semantic Systems, 20-22 September 2023, Leipzig, Germany* (Vol. 56, pp. 110–126). IOS Press. https://doi.org/10.3233/SSW230009

- Peer Reviewed
- Written in sole authorship
- Nominated for the best paper award of the SEMANTiCS 2023 R&I track
- My contributions to the paper:
  - Conception and writing of the paper (100%)
  - Review of related work (100%)
  - Improvement of the computation of semantic distances between entities through the additional consideration of entity descriptions (100%)
  - Generation of a representative dual-entity query dataset based on TREC datasets (100%)
  - Optimization of the pathfinding algorithm's hyperparameters α, β, and γ using the Simple optimizer (100%)
  - Full reimplementation of the pathfinding algorithm, now called BiPaSs (100%)
  - Evaluation of BiPaSs (100%)
  - Discussion of limitations and future work (100%)

Paper III

A Testbed for Dual-Entity Knowledge Panels

Martin, L., & Henrich, A. (2023a). A Testbed for Dual-Entity Knowledge Panels. In M. Leyer & J. Wichmann (Eds.), *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings, Marburg, Germany, October 9-11, 2023* (Vol. 3630, pp. 231–238). CEUR-WS.org. https://ceur-ws.org/Vol-3630/LWDA2023-paper21.pdf

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (90%)
  - Review of related work (90%)
  - Conception and implementation of a testbed facilitating the development and evaluation of dual-entity knowledge panels (100%)
  - Implementation of four initial presentation formats for entity relationship explanations in the testbed (100%)
  - Discussion of limitations and future work (80%)

Paper IV

On the Form of Research Publications for Use in Scientific Knowledge Graphs

Martin, L., Jegan, R., & Henrich, A. (2021). On the Form of Research Publications for Use in Scientific Knowledge Graphs. *Wissensorganisation 2021: 16. Tagung Der Deutschen Sektion Der Internationalen Gesellschaft Für Wissensorganisation (ISKO) (Wissorg'21)*, accepted for publication, preprint available. https://easychair.org/publications/preprint/SDQs

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (80%)
  - Review of related work (80%)
  - Investigation of available publication forms and their compatibility with scientific knowledge graphs (70%)
  - Discussion of future work (70%)

Since this paper has not been published yet, the following pages provide a preprint version, as required by the regulations.

On the Form of Research Publications for Use in Scientific Knowledge Graphs

Leon Martin
leon.martin@uni-bamberg.de
University of Bamberg
Bamberg, Bavaria, Germany

Robin Jegan
robin.jegan@uni-bamberg.de
University of Bamberg
Bamberg, Bavaria, Germany

Andreas Henrich
andreas.henrich@uni-bamberg.de
University of Bamberg
Bamberg, Bavaria, Germany

ABSTRACT

Current research proposes scientific knowledge graphs to support various research activities by acquiring and integrating scientific information including research publications. The accompanying envisaged shift towards knowledge graph based research motivates rethinking the form of research publications since the traditional form of research publications, i.e., self-contained documents, may leave opportunities unused. This paper investigates different publication forms that are used in scientific knowledge graphs, identifies their flaws from the authors', providers', and readers' perspectives subsequently, and finally outlines a first set of requirements that a publication form tailored for use in scientific knowledge graphs should fulfill.

KEYWORDS

scientific knowledge graphs, research publications, publication forms, requirements

1 INTRODUCTION

Motivated by the growing number of scientific communities and research publications [2], the interest in Scientific Knowledge Graphs (SciKGs) [11], also known as scholarly knowledge graphs [13, 14] or research knowledge graphs [8], i.e., knowledge graphs that acquire and integrate scientific information in a knowledge base [1, 4], is on the rise. In this regard, the TIB Leibniz Information Centre for Science and Technology and the L3S Research Centre in Hannover are important contributors. One of their recent papers [2] presents a thorough requirements analysis for their Open Research Knowledge Graph (ORKG) [8][1], an already operative scientific knowledge graph, that is intended to facilitate typical research activities like finding related work, assessing relevance, and reproducing results among others.

Nevertheless, for many decades, research publications have come in the form of self-contained documents, so-called papers. The shift away from document-centric research towards the envisaged knowledge graph based research, however, also includes reconsidering the form of research publications as traditional papers may not take full advantage of the new paradigm's opportunities. Therefore, this paper first investigates current publication forms that are used in SciKGs (section 2), identifies their flaws (section 3), and finally outlines requirements for a publication form tailored for use in SciKGs (section 4), before drawing a conclusion in section 5.

[1] https://www.orkg.org (accessed 19/10/2021)

2 CONTEXT & RELATED WORK

Knowledge graphs leverage the Resource Description Framework (RDF) to represent information as triples, each comprising a subject (an entity), a predicate (a property), and an object (an entity or a literal) [3]. A set of RDF triples span a graph, i.e., the RDF or knowledge graph. Ontologies, based on which knowledge graphs are constructed, formally define what entities mean in a given domain, what features they possess, and thereby how entities, properties—both identified via Internationalised Resource Identifiers (IRIs), a generalization of Uniform Resource Identifiers (URIs) [3]— and literals can be arranged in RDF triples [7, 17–25].

In the context of SciKGs, one key challenge is the integration of research publications, which carry a significant part of the available scientific knowledge. From the perspective of SciKG providers, the goal here is to achieve a high coverage of research publications and gather a large user base, whereas authors want to increase the visibility of their publications with little additional effort[2]. Since ontologies provide the formal base of SciKGs as well, they determine how research publications are integrated and thereby what publication forms are supported by the knowledge graph. At the same time, the standard publication form determines the ontology since providers of SciKGs want to maximize the number of potential contributors. As a result, ontologies of SciKGs and publication forms must evolve together due to their mutual influence.

Depending on the intended use cases, SciKGs' ontologies can be designed to focus on the representation of *contextual* information for describing research publications or even to allow the representation of their *contentual* information (cf. Figure 1), e.g., the publications' contributions[3]. For this purpose, the usage of knowledge graph cells [15] is an option. Representing contentual information, however, imposes various challenges like the expression of opinion forming [1] that have to be addressed in the future.

There are different forms of research publications that are already used or lend themselves to be used in SciKGs. The obvious option is to retain the traditional self-contained documents, herein called *document-based publications*. This approach provides the benefit that authors can prepare their publications in the way they are used to. Currently, the ORKG supports this publication form, i.e., document-based publications can be added via Digital Object Identifiers (DOIs) or manually to the knowledge graph. Following the addition, contributors can establish links to other entities in the knowledge graph and add the contributions provided in their publication as new entities based on templates.

[2] As prominently stated on the ORKG's homepage[1] and in [14], aspects like compliance to the FAIR data principles are also important for the success of a SciKG.

[3] cf. https://www.orkg.org/orkg/paper/R134713 (accessed 19/10/2021): A paper featuring multiple contributions that are explicitly represented in the ORKG.

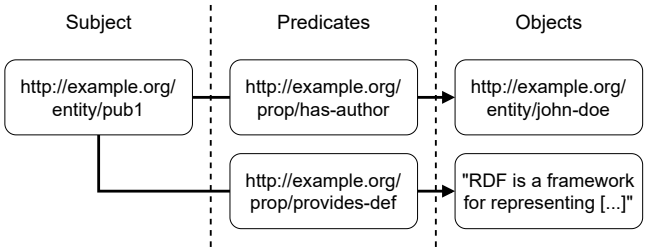


Figure 1: A simple exemplary knowledge graph consisting of two RDF triples. The upper triple provides contextual information, the lower triple contentual information of the publication *pub1*. All non-literal triple members are identified using IRIs.

For indexing and categorization purposes, many research organizations and publishers including the Association for Computing Machinery (ACM), Springer, and Elsevier recommend the usage of keywords in papers. However, these keywords can often be set arbitrarily by the authors, thus impeding their utility for the knowledge graph integration process due to the missing IRIs. To tackle this, more advanced means of classifying research publications are applied. For instance, the ACM promotes the usage of their ACM Computing Classification System (CCS), a poly-hierarchical ontology for classifying research publications in the computing domain, whose current version launched in 2012 [6]. One downside of the CCS is its reliance on proprietary identifiers instead of IRIs which would facilitate connecting to other RDF resources.

To simplify the integration process by closing the gap between document-based publications and knowledge graphs, one option is to leverage OpenIE [5] systems and knowledge graph construction [12] to transform text into knowledge graphs. For the scientific domain, [11] proposes *SciIE*, a unified framework for constructing SciKGs based on scientific literature. Herein, knowledge graphs that result from transforming document-based publications will be referred to as *RDF-transformed publications*. The problem is that knowledge graph construction relies on several to date error-prone Natural Language Processing (NLP) techniques like entity recognition, relation extraction, and coreference resolution, thus yielding sub-optimal results (cf. [11]). Approaches that employ neural networks are able to beat previous approaches (cf. [9]), but the results still remain below the quality required for SciKGs. However, the current interest in NLP and the further investigation of machine learning techniques in this field will eventually result in improved knowledge graph construction approaches, as well. Note that we assume in this paper that knowledge graph construction techniques are able to produce high quality RDF-transformed publications for a fair comparison.

In contrast to traditional document-based publications, another publication form has emerged in recent years, namely *nanopublications* [10]. This publication form consists of one "[...] atomic snippet of a formal statement [...]" [10] accompanied by the origin of this information, also mentioned as provenance, and metadata. The information is formatted as linked data, more precisely as RDF graphs, and thus far mostly used in the Life Science domain. Whereas traditional self-contained documents usually provide accompanying information to the subject such as sections on introduction, related work, and future work among others, nanopublications do not include such contextual information but use links instead.

3 PROBLEM IDENTIFICATION

To conceptualize the requirements of a future publication form tailored for use in SciKGs, it is necessary to first investigate the flaws of the previously described publication forms in this context. For this purpose, three perspectives are considered: The authors' perspective, which represents the group producing research publications. The providers' perspective, which represents the group maintaining the SciKG. The readers' perspective, which represents the group exploring the SciKG.

Authors' perspective. It is common that authors produce publications that overlap to a certain degree. For instance, a researcher in the field of linked data will produce multiple publications that rely on RDF as a concept, or a formal definition of knowledge graphs. For readability, the necessary concepts and definitions must be introduced to the readers in each publication. In the case of document-based publications, this results in passages—typically found in sections called *Related Works* or *Foundations*—that are effectively redundant across multiple papers even though they are often rephrased to avoid plagiarism. As a result, authors are forced to spend valuable time, effort, and space to produce passages that do not provide any contribution[4]. RDF-transformed publications do not mitigate this problem, as they rely on document-based publications prepared in the same manner.

Although nanopublications do not contain redundant information by definition, they require a different amount of time and effort from the author, since their formal nature necessitates the study of the guidelines[5]. The formal structure of nanopublications is a restricting factor in another way, meaning the ontology that is used as its basis. Authors without experience in nanopublications or who study in domains without any entities already present in a system based thereon, would need to construct the definitions, concepts, and other data relevant to their publication, thus creating an ontology on their own, before being able to publish a nanopublication.

Providers' perspective. The problems that arise from a providers' perspective are related to the integration of research publications into the knowledge graph. The integration of new publications requires the identification and addition of entities and relations that are not yet represented in the graph, e.g., contributions the publication provides, as well as the recognition and linkage of entities and relations that are already represented in the graph, e.g., an author who is already part of the graph. Document-based publications do not contain useful information to decide whether a mentioned entity or relation is present in the knowledge graph or not. Of course, many publications today include ORCID identifiers to disambiguate authors or DOI information to identify referenced documents, but they are just an intermediate step to obtain the actual IRIs of entities, which are required for the integration. As already discussed in Section 2, keywords and similar systems like

[4]Note that adjustments made to said concepts and definitions, or contextualizing information provided by the authors are not meant here as they represent actual contributions.

[5]http://nanopub.org/guidelines/working_draft (accessed 19/10/2021)

the CCS do not suffice for properly and conveniently categorizing document-based publications.

In contrast to document-based publications, RDF-transformed publications provide the benefit that mentioned entities are assigned their correct IRIs as long as the entity linking within the knowledge graph construction process succeeds. If necessary, new entities can be created and added to the SciKG, too. Relations are extracted as well such that RDF-transformed publications include them, thus further simplifying the integration process.

Similar to RDF-transformed publications, using nanopublications can improve the integration of new research publications into an existing knowledge graph. However, the infrastructure necessary for the formal setup of nanopublications would have to be offered, maintained and developed by the providers. Furthermore, the effort in establishing such an infrastructure including definitions and other information relating to a domain without prior usage of nanopublications would be substantial, especially when considering the training of authors that should use the system. Advantages, on the other hand, would be significant, enabling filtering according to certain subjects, authors or dates.

Readers' perspective. When readers view a research publication that interests them, they may want to investigate other research publications that are related to the topic. This includes publications that are referenced by the publication at hand as well as publications that reference the publication at hand. Document-based research publications only mention referenced publications, thus only supporting a backwards search. In contrast, a forwards search, i.e., the investigation of referencing publications, is only possible using external tools like Semantic Scholar[6].

RDF-transformed publications allow readers to both view the original document-based publication and the generated knowledge graph, given that a suitable knowledge graph visualization is available. The former provides the familiarity and readability of the traditional publication form, while the latter could be leveraged to implement features like a forwards search. However, readers have the additional effort of viewing two representations of the same publications for different use cases instead of one coherent representation, which is not optimal from a usability point of view.

To the best of our knowledge, a system displaying nanopublications with a mature interface does not exist yet, thus decreasing the usefulness for potential readers, since they would have to use scripts or APIs to query publications. Furthermore, citation figures are not available when compared to the other publication forms[7], which enables a quick estimation of how influential or popular a given paper is, thus presenting a simple filter for the reader.

4 REQUIREMENTS

As shown in Section 3, each publication form considered here has advantages as well as disadvantages regarding their use in SciKGs. Based on our findings, we propose a first set of requirements for a future publication form tailored for use in this context. As explained in Section 2, note that the ontology of the SciKG has to be designed in a way compatible to the future publication form.

[6]https://www.semanticscholar.org (accessed 19/10/2021)

[7]The need for new scholarly communication incentive measures that arises from the shift towards knowledge graph based research is also noted in [1].

Table 1 lists the requirements set; the description is given below. As one can see, all three perspectives are covered by some requirements with respect to the flaws we identified from each perspective. That being said, it is difficult to really provide a clear-cut mapping from the requirements to the affected groups since many requirements somewhat influence multiple perspectives as well as each other. Hence, we assigned each requirement to the groups affected the most.

Table 1: A first set of requirements for a future publication form tailored for use in scientific knowledge graphs. The three columns, (A)uthors', (P)roviders', (R)eaders', indicate the affected perspective(s).

#	Requirement	A	P	R
1	Preparation of main contributions in natural language	×		
2	Import of knowledge	×		
3	Markup of knowledge	×	×	
4	Provision of tooling for obtaining IRIs	×	×	
5	Enriched representation of publications at view time			×

Requirement 1 prescribes that authors shall be able to prepare their main contributions in natural language as they are used to in order to minimize training effort. The idea of nanopublications is interesting, but for a general domain SciKG the limited flexibility caused by the strict ontology does not suffice for properly expressing contributions in all scientific fields. However, for domain specific SciKGs with a narrow scope, e.g., a knowledge graph with focus on empirical studies in the Life Science domain, nanopublications may be a viable option.

Requirement 2 addresses the problem of redundancy across publications. To avoid redundant passages, authors shall be able to import, i.e., link, knowledge that is already present in the knowledge graph within their publications. For this purpose, they shall be able to specify the knowledge they rely on using IRIs where appropriate. For example, if authors rely on a certain definition for RDF that is already present in the SciKG, they shall be able to specify the definition's IRI in their publication. In combination with Requirement 5, this eliminates redundant passages across publications. Of course, changes made to, for example, imported definitions or contextualizing information still have to be explicitly provided by the authors.

Due to Requirement 1, the main part of the future publication form will be written in natural language. As a consequence, it is necessary to either mark up original entities and their relations manually or to leverage knowledge graph construction when an automated integration process is the goal. That being said, the best option may be to use both approaches. To this end, Requirement 3 states that knowledge graph construction techniques shall be applied first to generate suggestions for mentioned entities and their relations. Subsequently, authors shall review and adapt the suggestions if necessary. To mark up RDF elements that are already present in the SciKG, an RDFa[8]-like approach could be used.

[8]https://www.w3.org/TR/rdfa-primer (accessed 19/10/2021)

Despite the suggestions provided using knowledge graph construction, reviewing the suggestions and marking up their publications demands additional effort from the authors. Hence, tooling that facilitates activities like searching for IRIs of mentioned entities shall be provided, which represents Requirement 4.

From compliance with Requirement 2, issues regarding the readability arise since IRIs replace actual text written in natural language. To tackle this, we can make use of the capabilities SciKGs provide: At view time, readers shall be provided a single document-based representation of the publication that is enriched using information currently available in the knowledge graph, constituting Requirement 5. The generated version shall comprise the original text by the authors as well as the natural language fragments that are generated by resolving the links to imported knowledge. To enable backwards and forwards search, clickable IRIs to both referenced publications and referencing publications shall be provided. Depending on the information that is represented in the respective SciKG, other information could be included as well, which is worth exploring.

In summary, the goal is to retain the familiarity of document-based publications while exploiting the opportunities that arise from the use of SciKGs with small additional effort. Requirements 2 and 5 represent a positive aspect in this regard, as they allow saving resources that would otherwise be spent on producing redundant passages. In return, the markup process implies additional effort, thus representing an important aspect for future work.

5 CONCLUSION

This paper provided an overview of publication forms used in SciKGs and subsequently described flaws that arise from their use in this context. Then, we outlined a first set of requirements that a publication form tailored for use in SciKGs should fulfill. The next steps are to refine the requirement set by further investigating the use cases and to design a suitable publication form. Regarding the latter, we are investigating custom commands for LaTeX documents for importing knowledge via IRIs and a preprocessing step for their resolution. In a sense, this approach is similar to the famous Project Xanadu[9] which proposes a transclusion mechanism to include parts of documents in other documents.

REFERENCES

[1] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria-Esther Vidal. 2018. 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*. ACM, 1:1–1:6. https://doi.org/10.1145/3227609.3227689

[2] Arthur Brack, Anett Hoppe, Markus Stocker, Sören Auer, and Ralph Ewerth. 2020. Requirements Analysis for an Open Research Knowledge Graph. 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 (Lecture Notes in Computer Science, Vol. 12246)*. Springer, 3–18. https://doi.org/10.1007/978-3-030-54956-5_1

[3] Richard Cyganiak, David Hyland-Wood, and Markus Lanthaler. 2014. RDF 1.1 Concepts and Abstract Syntax. (01 2014). Retrieved July 28, 2021 from https://www.w3.org/TR/rdf11-concepts

[4] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a Definition of Knowledge Graphs. In *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016 (CEUR Workshop Proceedings, Vol. 1695)*. CEUR-WS.org. http://ceur-ws.org/Vol-1695/paper4.pdf

[5] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM* 51, 12 (2008), 68–74. https://doi.org/10.1145/1409360.1409378

[6] Association for Computing Machinery (ACM). 2021. *Computing Classification System*. Association for Computing Machinery (ACM). Retrieved July 28, 2021 from https://dl.acm.org/ccs

[7] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2020. Knowledge Graphs. *CoRR* (2020). https://arxiv.org/abs/2003.02320

[8] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*. ACM, 243–246. https://doi.org/10.1145/3360901.3364435

[9] Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V. Chawla, and Meng Jiang. 2019. The Role of "Condition": A Novel Scientific Knowledge Graph Representation and Construction Model. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 1634–1642. https://doi.org/10.1145/3292500.3330942

[10] Tobias Kuhn, Albert Meroño-Peñuela, Alexander Malic, Jorrit H. Poelen, Allen H. Hurlbert, Emilio Centeno, Laura I. Furlong, Núria Queralt-Rosinach, Christine Chichester, Juan M. Banda, Egon L. Willighagen, Friederike Ehrhart, Chris T. A. Evelo, Tareq B. Malas, and Michel Dumontier. 2018. Nanopublications: A Growing Resource of Provenance-Centric Scientific Linked Data. *CoRR* (2018). http://arxiv.org/abs/1809.06532

[11] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 3219–3232. https://doi.org/10.18653/v1/d18-1360

[12] José-Lázaro Martínez-Rodríguez, Ivan López-Arévalo, and Ana B. Ríos-Alvarado. 2018. OpenIE-based approach for Knowledge Graph construction from text. *Expert Syst. Appl.* 113 (2018), 339–355. https://doi.org/10.1016/j.eswa.2018.07.017

[13] Allard Oelen, Mohamad Yaser Jaradeh, Kheir Eddine Farfar, Markus Stocker, and Sören Auer. 2019. Comparing Research Contributions in a Scholarly Knowledge Graph. In *Proceedings of the Third International Workshop on Capturing Scientific Knowledge co-located with the 10th International Conference on Knowledge Capture (K-CAP 2019), Marina del Rey, California , November 19th, 2019 (CEUR Workshop Proceedings, Vol. 2526)*. CEUR-WS.org, 21–26.

[14] Allard Oelen, Mohamad Yaser Jaradeh, Markus Stocker, and Sören Auer. 2020. Generate FAIR Literature Surveys with Scholarly Knowledge Graphs. In *JCDL '20: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, Virtual Event, China, August 1-5, 2020*. ACM, 97–106. https://doi.org/10.1145/3383583.3398520

[15] Lars Vogt, Jennifer D'Souza, Markus Stocker, and Sören Auer. 2020. Toward Representing Research Contributions in Scholarly Knowledge Graphs Using Knowledge Graph Cells. In *JCDL '20: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, Virtual Event, China, August 1-5, 2020*. ACM, 107–116. https://doi.org/10.1145/3383583.3398530

[9] https://xanadu.com (accessed 19/10/2021)

Paper V

RDFtex:
Knowledge Exchange Between LaTeX-Based Research Publications and Scientific Knowledge Graphs

Martin, L., & Henrich, A. (2022a). RDFtex: Knowledge Exchange Between LaTeX-Based Research Publications and Scientific Knowledge Graphs. In G. Silvello, Ó. Corcho, P. Manghi, G. M. D. Nunzio, K. Golub, N. Ferro, & A. Poggi (Eds.), *Linking Theory and Practice of Digital Libraries - 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20-23, 2022, Proceedings* (Vol. 13541, pp. 26–38). Springer. https://doi.org/10.1007/978-3-031-16802-4_3

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (80%)
  - Review of related work (90%)
  - Conception of RDFtex, a framework facilitating the import of research contributions from scientific knowledge graphs and the export of research contributions to scientific knowledge graphs via four custom LaTeX commands (90%)
  - Implementation of the RDFtex framework (100%)
  - Implementation of a makeshift scientific knowledge graph called MinSKG for showcasing RDFtex (100%)
  - Preliminary evaluation of RDFtex including conducting a qualitative user study with three test persons (100%)
  - Discussion of limitations and future work (90%)

Paper VI

RDFtex in-depth:
knowledge exchange between LATEX-based research publications and Scientific Knowledge Graphs

Martin, L., & Henrich, A. (2023b). RDFtex in-depth: knowledge exchange between LATEX-based research publications and Scientific Knowledge Graphs. *IJDL: International Journal on Digital Libraries*, 1–19. https://doi.org/10.1007/s00799-023-00370-5

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (90%)
  - Review of related work (100%)
  - Further development of the RDFtex framework (100%)
  - Proposal of a research and publication process involving RDFtex (90%)
  - Investigation of problems associated with the practical utilization of RDFtex such as the identification of IRIs (90%)
  - Evaluation of RDFtex including conducting a qualitative user study with ten test persons with varied academic backgrounds and LaTeX proficiency levels (100%)
  - Discussion of limitations and future work (90%)

Paper VII

Specification and Validation of Quality Criteria for Git Repositories using RDF and SHACL

Martin, L., & Henrich, A. (2022b). Specification and Validation of Quality Criteria for Git Repositories using RDF and SHACL. In P. Reuss, V. Eisenstadt, J. M. Schönborn, & J. Schäfer (Eds.), *Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, Hildesheim (Germany), Oktober 5-7th, 2022* (Vol. 3341, pp. 124–135). CEUR-WS.org. https://ceur-ws.org/Vol-3341/WM-LWDA_2022_CRC_1149.pdf

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (90%)
  - Review of related work (90%)
  - Conception of QuaRe, a tool facilitating the validation of GitHub repositories against sets of quality constraints that correspond to the requirements of the respective project types (90%)
  - Conception of two representative project types comprising eleven and four quality constraints, respectively (90%)
  - Implementation of QuaRe and its validation functionality using two alternative approaches, one based on SHACL and another one based on OWL (100%)
  - Evaluation of QuaRe using twenty trending GitHub repositories (100%)
  - Discussion of limitations and future work (80%)

Paper VIII

Assessing the FAIRness of Software Repositories using RDF and SHACL

Hummel, T., Martin, L., & Henrich, A. (2024). Assessing the FAIRness of Software Repositories Using RDF and SHACL. In A. A. Salatino, M. Alam, F. Ongenae, S. Vahdati, A. L. Gentile, T. Pellegrini, & S. Jiang (Eds.), *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands* (Vol. 60, pp. 160–175). IOS Press. https://doi.org/10.3233/SSW240014

- Peer Reviewed
- My contributions to the paper:
  - Conception and writing of the paper (50%)
  - Review of related work (50%)
  - Conception of a new project type facilitating the validation of GitHub repositories against FAIR best practices (50%)
  - Full redesign of the repository representations (50%)
  - Implementation of the QuaRe tool extensions (30%)
  - Evaluation of QuaRe based on six GitHub repositories expected to be FAIR and 217 popular GitHub repositories (70%)
  - Discussion of limitations and future work (70%)