GESELLSCHAFT
FÜR INFORMATIK

Carsten Binnig, Andreas Henrich,
Daniela Nicklas, Maximilian E. Schüle, Klaus Meyer-Wegener
(Hrsg.)

Datenbanksysteme für Business, Technologie und Web (BTW 2025)

Workshopband

03. – 07. März 2025
Bamberg, Deutschland

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-363

ISSN 2944-7682 (Online)

Volume Editors

Prof. Dr. Carsten Binnig
TU Darmstadt
Lehrstuhl für Data Engineering
Hochschulstr. 10, 64289 Darmstadt
carsten.binnig@cs.tu-darmstadt.de

Prof. Dr. Andreas Henrich
Otto-Friedrich-Universität Bamberg
Lehrstuhl für Medieninformatik
An der Weberei 5, 96047 Bamberg
andreas.henrich@uni-bamberg.de

Prof. Dr. Daniela Nicklas
Otto-Friedrich-Universität Bamberg
Lehrstuhl für Informatik, insbesondere Mobile Softwaresysteme/Mobilität
An der Weberei 5, 96047 Bamberg
daniela.nicklas@uni-bamberg.de

Prof. Dr. Maximilian E. Schüle
Otto-Friedrich-Universität Bamberg
Juniorprofessur für Informatik, insb. Data Engineering
An der Weberei 5, 96047 Bamberg
maximilian.schuele@uni-bamberg.de

Prof. i. R. Dr. Klaus Meyer-Wegener
Friedrich-Alexander Universität Erlangen-Nürnberg
Lehrstuhl für Informatik 6 (Datenmanagement)
Martensstraße 3, 91058 Erlangen
klaus.meyer-wegener@fau.de

Vorwort

Die 21. Fachtagung “Datenbanksysteme für Business, Technologie und Web” (BTW‘25) des Fachbereiches “Datenbanken und Informationssysteme” (DBIS) der Gesellschaft für Informatik (GI) findet vom 3. bis 7. März 2025 an der Otto-Friedrich-Universität Bamberg statt (https://btw2025.gi.de/).

Die BTW‘25 setzt damit die lange Tradition dieser Konferenzreihe fort, seit mehr als 40 Jahren ist sie die größte Datenbankkonferenz im deutschsprachigen Raum und damit das größte Treffen von Wissenschaftlerinnen und Wissenschaftler, die im Datenbankbereich arbeiten, Entwicklerinnen und Entwicklern aus der Industrie und Studierenden, die sich in diese Richtung spezialisieren. Die BTW 2025 ermöglicht allen den fachlichen Austausch zu aktuellen Forschungs- und Entwicklungsthemen und -ergebnissen.

Die Geschichte der Universität Bamberg geht auf das Jahr 1647 zurück, als die Academia Bambergensis gegründet wurde, die sich zunächst auf Philosophie und Theologie konzentrierte. Im Jahr 1773 wurde sie zur Universitas Ottoniano-Fridericiana erweitert und bot auch Rechtswissenschaften und Medizin an. Nach der Säkularisierung im Jahr 1803 wurde die Universität jedoch auf ein Lyzeum reduziert und verlor ihre juristischen und medizinischen Fakultäten. Im Jahr 1923 wurde die Einrichtung in eine Philosophisch-Theologische Hochschule umgewandelt, die weiterhin auf Philosophie und Theologie ausgerichtet war. Parallel dazu wurde zwischen 1958 und 1972 die Pädagogische Hochschule als eigenständige Einrichtung der Universität Würzburg tätig. Beide Einrichtungen wurden 1972 zur Gesamthochschule vereinigt. Dieser Prozess endete 1979 mit der offiziellen (Wieder-)Erlangung des Status einer Universität. Weitere Fortschritte folgten mit der Gründung der Fakultät für Wirtschaftsinformatik und Angewandte Informatik (WIAI) im Oktober 2001. Im August 2012, expandierte die Universität mit der Eröffnung eines neuen Gebäudes, in dem die BTW 2025 stattfindet.

Die Data Science Challenge, zwei Tutorials und Industriebeiträge komplettieren das Programm der BTW‘25. Ergänzt wird das wissenschaftliche BTW-Programm durch fünf im Vorfeld stattfindende Workshops, die aktuelle Forschungstrends aufgreifen und auf denen technische Beiträge sowie Vision Paper zu diesen vorgestellt und diskutiert werden. Diese Beiträge sind in diesem Workshop-Tagungsband enthalten:

- Workshop on Novel Data Management Ideas on Heterogeneous Hardware Architectures (NoDMC)
- Workshop on Big (and Small) Data in Science and Humanities (BigDS)
- Workshop on Data Engineering for Data Science (DE4DS)
- A Tutorial Workshop on ML for Systems and Systems for ML
- Workshop on Advances in Cloud Data Management (ACloudDM)
- Early Career Researcher Workshop
- Symposium on Data Driven Smart City Science and Transferability (DaSCiT)

Ein ganz besonderer Dank geht natürlich an die Vielzahl der Sponsoren, ohne deren Engagement die Durchführung einer BTW nicht möglich wäre. Auch gilt ein großer Dank der GI-Geschäftsstelle für die umfangreiche Unterstützung in der finanziellen Abwicklung der Tagung.

Vielen Dank an alle Beteiligten für das große Engagement!

Bamberg, im Februar 2025

Andreas Henrich, Daniela Nicklas, Maximilian E. Schüle (Universität Bamberg)

Tagungsleitung

Carsten Binnig (TU Darmstadt)

Leitung Workshop- und Tutorialprogramm

Klaus Meyer-Wegener (FAU Erlangen-Nürnberg)

Ehrenvorsitzender

Sponsoren

Wir danken den folgenden Unternehmen und Institutionen für die Unterstützung der Konferenz.s

Platin

Gold

Silber

Bronze

Inhaltsverzeichnis

Symposium on Data Driven Smart City Science and Transferability

Workshop on Advances in Cloud Data Management

Workshop on Big (and Small) Data in Science and Humanities

Workshop on Data Engineering for Data Science

Workshop on ML4Sys and Sys4ML

Workshop on Novel Data Management Ideas on Heterogeneous Hardware Architectures

Early Career Researcher Workshop

Data Science Challenge

Autorenverzeichnis

Symposium on Data Driven Smart City Science and Transferability

Commercialization for Public Value - A Path for Civic Technology Innovation to Diffuse Into Public Administrations

Luca T. Bauer [1]

Abstract: This paper explores the dynamics of commercialization as a strategy for civic technology (CivicTech) ventures to gain trust and legitimacy in public administration. CivicTech, rooted in grassroots innovation, often struggles with scalability and sustainability due to limited financial resources. In response, some ventures transition into government technology (GovTech), adopting professional and commercial practices. Through a mixed-methods study, including desk research and expert interviews, an examination is presented on how isomorphism, conoercive, mimetic, and normative, drives this transition. Findings highlight how commercialization fosters innovation diffusion into public sectors, balancing grassroots ideals with institutional expectations to enhance public value in smart cities and beyond.

Keywords: E-Government, Smart City, CivicTech, GovTech, Innovation Diffusion

1 Introduction

The rise of challenges in maintaining digital sovereignty [Fl20], coupled with public administrations' persistent inability to meet citizens' needs [HT22], poses a significant threat to democratic values. As the complexity of societal problems grows, public institutions are increasingly constrained by limited human capital, tight budgets, and inadequate digital infrastructure [DT24; Ni11]. To bridge these gaps, public organizations frequently turn to external solution providers, seeking innovative technologies to modernize their services and address critical issues, especially in smart city contexts [SKT22; Wi24].

Civic technology organizations (CivicTechs) have emerged as relevant players in this landscape, developing bottom-up driven technology solutions with and for public sector clients [To13], [Sa19]. Unlike traditional commercial enterprises, CivicTechs are typically motivated by public interest rather than profit, often embodying social entrepreneurial ideals. One prominent example is *Code for America*, which exemplifies the potential of grassroots innovation to solve pressing public problems [Sc16]. However, despite their admirable mission, many CivicTech ventures face challenges related to the sustainability and scalability of their solutions [YT21]. Public servants often view their non-commercial status skeptically, questioning their capacity for long-term viability and institutional trustworthiness [CD21].

In response to these limitations, a notable trend of the 2020s has been the rise of GovTech, an umbrella term combining "government" and "technology" [Bh22; NK24]. GovTech

[1] University of Bremen, Digital Public, Bibliothekstraße 1, 28359 Bremen, Freie Hansestadt Bremen, bauerl@uni-bremen.de, https://orcid.org/0009-0003-6140-9918

doi:10.18420/BTW2025-101

refers to small and medium enterprises (SMEs) and startups that specialize in developing technology for public sector clients [Bh22; Eu24]. These ventures vary widely, ranging from businesses that see governments as reliable, long-term clients to those focused on creating scalable solutions for smart cities and regions. Importantly, GovTech also includes socially motivated startups that seek to address public challenges through technological innovation [JR14].

Interestingly, some GovTech ventures originate as CivicTech initiatives, yet they deliberately choose to commercialize their operations [Bh22; Zh17], meaning shifting their strategy from non-profit and cost-covering to running principally for monetary gain [KP16]. This raises intriguing questions: could this shift from CivicTech to GovTech represent an evolutionary step aimed at gaining trust from public administrations? Evidence suggests that public servants are more likely to procure solutions from commercial ventures than from social initiatives, perceiving the former as more reliable and professional [Ni10].

We hypothesize that this is indeed the case. Many GovTech ventures do not pursue commercialization as an end in itself, instead, they utilize it as a strategic mechanism to align with public sector expectations [Ni10]. Through processes of isomorphic change, these organizations adapt their structures and practices to mirror those of traditional commercial entities, thereby enhancing their legitimacy and trustworthiness [DP83].

To explore this phenomenon, this paper addresses the following research question (RQ):

Do civic technology ventures commercialize their organization to attain public administrations' trust through mechanisms of isomorphic change?

In answering this question, the paper introduces the concepts of GovTech and CivicTech, situating them within the broader framework of public sector innovation and institutional isomorphism. Data from a structured literature review and expert interviews are used to provide empirical insights and to analyze the dynamics of commercialization as a trust-building strategy.

2 Theoretical Background

2.1 Civic and Government Technology

The last years have seen a rise in public, political, and academic debate on new forms of public-private partnerships and ventures as a solution to supply technology to public administrations [Bh22], [BJ24]. This is most visible in the more American phenomena of CivicTech and the European driven phenomenon of GovTech [Me22; NK24; Sc16]. Both signify channels for supplying technological innovation into the public sector, but at least in recent debate, are based on differing grounds.

Civic Technology (CivicTech) initiatives are rooted in grassroots, citizen-driven approaches, focusing on creating technology solutions that enhance civic engagement and improve the relationship between citizens and government institutions [To13; YT21]. These initiatives prioritize social impact over profitability, aligning closely with principles of social entrepreneurship. However, they often struggle with scalability and sustainability due to limited financial and structural resources [SVB09].

In contrast, Government Technology (GovTech) represents solutions supplied to the public sector by startups and SMEs, bridging the gap between innovation and financial viability [BJ24], [Eu24]. GovTech ventures retain a focus on public sector challenges but integrate business principles, enabling scalability and long-term impact [NK24]. They're often times seen as an alternative to bigtech, enabling a certain degree of digital infrastructure independence, but also creating new governance questions on economic development, public administrations absorption capacity and more [BJ24].

2.2 Public Sector Innovation

Public sector innovation refers to the process of creating and implementing new ideas, technologies, or practices to improve public services, governance, or organizational efficiency [CTS19]. Unlike innovation in the private sector, which is typically driven by competition and profit motives, public sector innovation often prioritizes social outcomes and equity, showing parallels between public and social entrepreneurship [Kl10; Ph15]. This distinction introduces unique challenges, such as the need to balance accountability and transparency with experimentation and risk-taking [KM18]. Bureaucratic inertia, limited resources, and fragmented decision-making processes further complicate the innovation landscape in public organizations, especially visible in regard to deeptech smart city needs [FSP20; Ne19].

At the same time, the public sector's role as a steward of societal progress makes innovation essential. Issues such as climate change, digital inequality, and aging populations demand creative and adaptive solutions [Ki15]. Governments are increasingly looking to leverage technology, data, and partnerships to modernize their operations. This is especially visible in the case of smart cities reacting to citizens demands and especially environmental challenges [Ac24; Ki22]. However, advancements often rely on external expertise, highlighting the importance of public-private partnerships with innovative ventures like CivicTech and GovTech [BJ24].

2.3 Diffusion of Innovation Theory

The public sector only seldomly innovates for and by itself. Even though scholars have shown in the last few years, that the state holds a central role in creating technological innovation, the first use and product development happens in and for private markets [Ma11].

This turns public sector innovation into a question of public-private partnership driven diffusion of innovation.

Diffusion of innovation (DOI) originally goes back to the idea that diffusion occurs through a process by which an innovation is communicated over time among members of a community, ultimately resulting in adoption or rejection [RSQ14]. The theory identifies five adopter categories - innovators, early adopters, early majority, late majority, and laggards - each playing a distinct role in the adoption process. The theory outlines several key factors that influence the likelihood and speed of innovation adoption: *relative advantage, compatibility, and observability* [RSQ14]. Relative advantage refers to the perceived improvement an innovation offers compared to existing solutions. Compatibility measures how well an innovation aligns with existing values, practices, and infrastructure. Lastly observability refers to the extent to which the benefits of an innovation are visible and demonstrable to potential adopters. These three pillars together can very well determine the likeliness of innovation adoption in smart cities, but in their theoretical foundations are mainly geared at private market actors, not covering public organizations and institutions.

2.4 Isomorphism

Building on the challenges of innovation diffusion in the public sector, institutional isomorphism provides a framework to understand how CivicTech ventures adapt to gain trust and legitimacy within bureaucratic environments. Isomorphism describes how organizations become similar through shared pressures, which fall into three categories: *coercive, mimetic, and normative* [DP83]. Coercive pressures stem from external mandates like legal requirements or regulatory standards. Mimetic pressures arise in uncertain environments, where organizations imitate successful counterparts to gain credibility. Normative pressures are shaped by professional norms, certifications, and industry best practices [LC20; Ni10].

3 Methodology

To investigate the dynamics of transitioning between CivicTech and GovTech and their implications for public sector innovation, this study employed a mixed-methods approach encompassing desk research and expert interviews [Ka21; VBB13; ZN16].

The desk research component involved mapping the German GovTech landscape through publicly accessible platforms. This included analyzing directories and data sources at municipal, federal state, and national levels. Additionally, recommendations from GovTech founders and ecosystem partners were incorporated to ensure a comprehensive inventory. This process resulted in a catalog of GovTech 256 ventures, which were subsequently differentiated based on the solutions they offered.

The study is part of a broader investigation into critical success factors for GovTech startups. A key element of this larger effort was conducting semi-structured interviews with over 100

GovTech founders in Germany. These interviews provided rich qualitative data, focusing on the development of the GovTech ecosystem and the challenges faced by public sector technology ventures.

For this paper particular attention was given to six not-for-profit ventures, that originated from civic projects before transitioning into the GovTech space. These cases can serve to highlight the unique trajectories of CivicTech initiatives evolving into commercially oriented GovTech ventures. The semi-structured nature of the interviews allowed for a flexible exploration of individual founder experiences, emphasizing the role

4 Results

4.1 Information Basis

The results of this study depict a partial quantity of the German GovTech ecosystem as a whole. We chose to focus on GovTechs as startups out of pragmatic reasons. Older ventures may lack the recent need for professionalization, as they have already become established, while including all civic initiatives becomes a scoping problem, opening the study to more than one hundred possible projects stemming from *ProjectTogether* events alone.

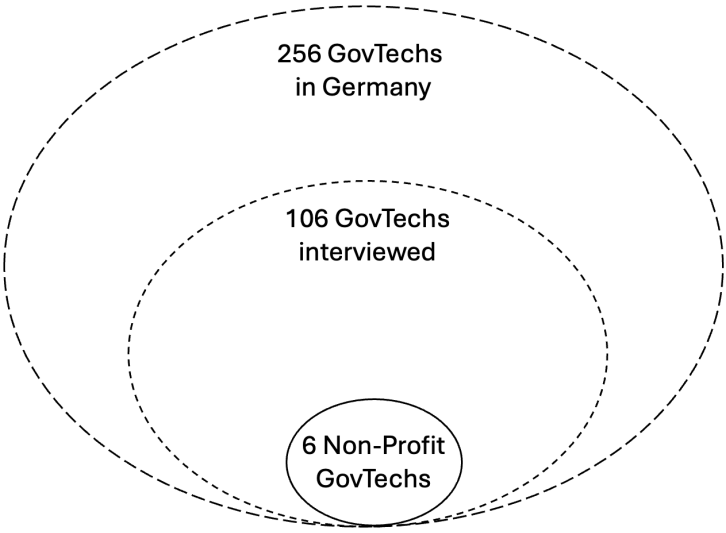


Fig. 1: Sample Civic-/GovTech

Through desk research, we assessed among others all federal state, country wide and European platforms covering the German landscape. This resulted in 207 GovTechs, which

were enriched by another 49 through personal recommendations and self representation on topic centered German events. Of these 256 identified ventures, 106 responded positively to interview request, of which six self-described as non-profit, not-for-profit, NGO, or charitable GovTechs, as visible in Figure 1. The ventures names will not be disclosed for reasons of confidentiality.

Further research has already shown that GovTech solutions [KS22], specifically in the German public sector, are made up of more than fifty percent offerings tailored to digital citizen services and smart city, which further are often intertwined, see Figure 2.

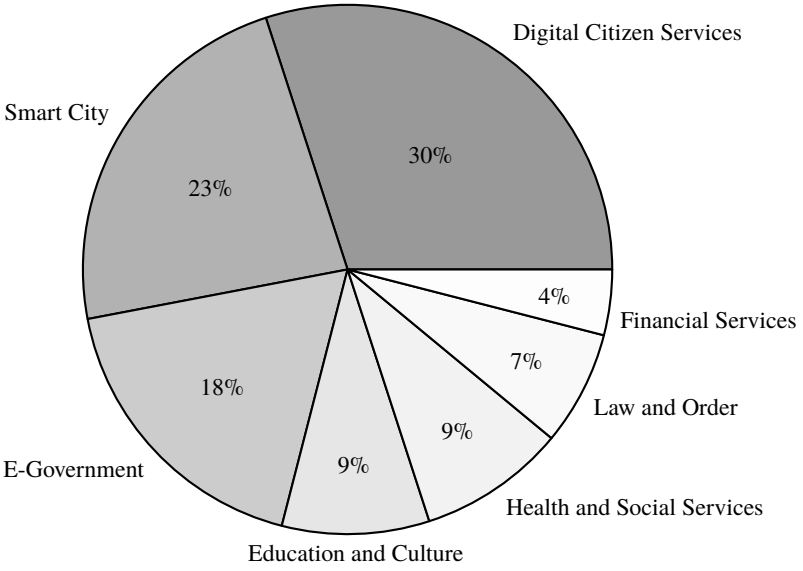


Fig. 2: Share of Solution Types Provided

4.2 Isomorphic Mechanisms

The 6 GovTech ventures, selected as above, self-described as not-for-profit. Nevertheless their shared identification, their legal makeup varies. Three ventures are organized as gGmbHs, two ventures take the form of UGs and one is organized as a registered organization (e.V.). Accordingly, simply applying organizational, legal makeup falls short for identifying professionalization [Ni10].

To solve the before detailed limitation in quantity, we employ exploratory qualitative interviews as main source of information. They serve to gain expert domain knowledge not accessible through other methods [Ka21]. For this purpose, the founders motivation and communicated factors relevant to their success were matched to the three forms of isomorphism: coercive, mimetic, and normative.

Citations given in the following have been translated and linguistically smoothed, but retain the contents as given by interview partners representing Civic-/GovTechs.

Interviewee	Quote
I	*"In 2009, when the association was founded, that [creating a commercial venture] was not yet the plan. And in 2010 [...] it was a great proof of concept to show that software platforms like ours could be used to support the Enquete Commission."*
II	*"The GovTechs I have met all make their results available so that others can also learn, which enables scaling. Most founders want to create value for everyone."*
III	*"The idea actually came from the WirVsVirus hackathon, [...] because we said we had to pick up the people who don't have access to the Internet."*
IV	*"In principle, the idea came about as part of the Münsterhack, [...] in which ideas that could improve the city of Münster in particular, but also society in general, were put forward."*
V	*./.*
VI	*"[...] The need that we had [...] was explicitly outlined to us by the city and they said that if you were to do something about it, then we would definitely be involved as a city."*

Tab. 1: Venture Creation Motivation

Venture creation motivation is a critical factor in illustrating the isomorphic processes that not-for-profit Civic-/GovTechs undergo when adapting to public sector environments. The motivations behind the formation of these ventures reflect how external pressures, both from societal needs and institutional demands, shape their trajectories [DP83]. Aligning with institutional expectations to build trust and legitimacy in the public sector, demonstrates the influence of coercive, mimetic, and normative isomorphism on their evolution.

The quotes in Table 1 provide insight into the diverse motivations driving the creation of not-for-profit GovTechs. Interviewee I highlights an early validation of their platform's utility in supporting government processes, underscoring the strategic alignment with institutional needs—a clear example of coercive isomorphism. Similarly, Interviewee VI describes how their venture responded to an explicit request from the city, emphasizing a direct, externally driven impetus for their creation. This further showcases how ventures align their missions with the expectations of public institutions to establish credibility. Other quotes reveal motivations rooted in societal impact, resonating with normative isomorphism. For instance, Interviewee II emphasizes value creation and knowledge sharing as a scaling mechanism, reflecting a commitment to shared public goals. Interviewees III and IV reference grassroots initiatives like hackathons, which foster innovation aimed at addressing local challenges. These origins illustrate how social norms and values shape their missions, reinforcing their legitimacy within the ecosystems.

Interviewee	Quote
I	*"We have been trying for a long time to get an external research group to conduct a larger study on the effect of Integrate as our main product."*
II	*"There are so many initiatives and funding programs in our country that you can't see the wood for the trees. They all want something different, which is why we have found that you have the most success if you focus locally on a region and adapt to its requirements."*
III	*"We received prize money as part of the Münsterhack. [...] We received 1,500€ in the Solution Enabler Program, which follows on from the Münsterhack, but we had to continue to work more professionally."*
IV	*./.*
V	*"So it is often the case that people [...] who play through the same problem, the solutions are good, we are actually relatively well processed, etc., but they don't know which demands they don't match."*
VI	*"We have been working with municipalities for a long time. At a certain point, their own inability forced us to organize planning processes and pricing models of other, similar companies so that we could compensate for their problems and continue to have them as clients."*

Tab. 2: Coercive Isomorphism

Coercive isomorphism refers to the pressures that organizations face from external forces such as regulations, funding mechanisms, or expectations imposed by dominant stakeholders [DP83]. These pressures often compel organizations to adapt their structures, strategies, or practices to align with the demands of powerful actors, ensuring their survival and legitimacy within the environment they operate. For not-for-profit GovTechs, coercive isomorphism is particularly significant as they navigate the institutional complexities of public sector collaboration and funding.

The quotes in Table 2 provide evidence of coercive isomorphism shaping the practices of not-for-profit GovTechs. Interviewee I highlights an effort to secure external validation through a formal study on their product's effectiveness. This reflects a pressure to meet evidence-based expectations from stakeholders such as public administrations or founders, demonstrating the influence of coercive forces on their operational focus. Interviewee II underscores the challenge of navigating a fragmented funding landscape, noting the need to adapt to local regional requirements for success. This indicates how funding conditions and regional demands act as coercive pressures, shaping the venture's strategic priorities. Similarly, Interviewee III describes the necessity of professionalization after receiving structured funding through programs like the Münsterhack. The structured nature of these programs imposes expectations of maturity and operational standards on the venture. Further, Interviewee VI illustrates how municipalities' challenges forced the venture to replicate planning and pricing models from similar companies to maintain their client base. This represents coercive pressure exerted by the operational inadequacies of public sector clients, pushing the GovTech to align its practices with more conventional market-oriented standards.

Interviewee	Quote
I	*"We actually come from a background of activism and fundraising, which is why it was difficult for us to win contracts. Other companies [GovTechs] don't work in the public sector with cold calling, so we gave that up for us as well and focused on relationship organizing.*"
II	./.
III	./.
IV	./.
V	*"Due to the difficult tendering situation, we now prefer to take on "piggyback" contracts, i.e. work on behalf of other organizations or departments that receive contracts from public organizations."*
VI	./.

Tab. 3: Mimetic Isomorphism

Mimetic isomorphism arises in situations of uncertainty, where organizations imitate the practices of more established or successful peers to gain legitimacy and credibility [DP83]. This type of isomorphism is particularly prevalent when organizations operate in uncertain or competitive environments, such as not-for-profit GovTechs navigating public sector ecosystems. Mimetic adaptation often involves replicating strategies or structures that are perceived to be effective, aligning the organization with recognized standards or practices.

The quotes in Table 3 illustrate instances of mimetic isomorphism among not-for-profit GovTechs. Interviewee I describes a shift from activism-oriented methods, such as fundraising and cold calling, to relationship organizing. This change reflects an imitation of successful strategies employed by other GovTechs, which better align with the expectations of public sector clients. The adaptation underscores how uncertainty in acquiring contracts pushes organizations to replicate practices seen as effective within their domain. Similarly, Interviewee V highlights the use of "*piggyback*"contracts, where the venture collaborates with other organizations or departments already engaged in public sector contracts. This approach reflects a mimetic adaptation to the challenges posed by complex tendering processes. By aligning their operations with established public sector partners, these GovTechs reduce risk and enhance their ability to deliver services effectively.

It is important to note that these findings are based on ventures that have not or will not fully transition into commercial entities. As such, they represent a subset of the GovTech landscape, focusing on ventures that continue to operate within a not-for-profit framework. Further exploration of commercialized GovTechs is necessary to fully understand how mimetic isomorphism operates across the market. This limitation highlights the need for broader research, which is discussed further in the limitations section of the paper.

Interviewee	Quote
I	*"We have to make open source software when we make software, and the software we release should be as easy to use as possible for anyone who wants to use it [which is in line with public demanded public money - public code policies]."*
II	*./.*
III	*./.*
IV	*"At the moment, we're developing it without any income anyway and I'd say it's important to us to educate people about it and we're doing it on a voluntary basis in that sense [attaining social legitimacy]."*
V	*./.*
VI	*"And only then [after a lot of trial and error working purely non-profit] did the legal entity follow. [...] So through the then eighth customer of our eighth partner, the city of Regensburg, who wanted a contract [and had to create the venture]."*

Tab. 4: Normative Isomorphism

Normative isomorphism arises from professionalization and the establishment of shared norms, values, and practices within a field [DP83]. This type of isomorphism is often driven by education, professional networks, and industry standards, leading organizations to conform to widely accepted practices and ethical frameworks. For not-for-profit GovTechs, normative isomorphism manifests in their alignment with social and professional expectations, such as public benefit, transparency, and accountability, as before mentioned in venture motivation.

The quotes in Table 4 reflect normative isomorphism in action. Interviewee I emphasizes the commitment to creating open-source software that aligns with public policies advocating for "*public money, public code*". This adherence to ethical norms and public sector values highlights how GovTechs align their practices with the expectations of their domain, reinforcing their legitimacy as socially responsible entities. Interviewee IV illustrates a normative focus on voluntary efforts and public education, aiming to attain social legitimacy. By prioritizing education and voluntary contributions, the venture embodies the professional and social norms of civic responsibility and inclusivity, central to their mission-driven objectives. Interviewee VI describes how their purely nonprofit origins eventually led to the formation of a legal entity in response to a public sector client's requirement. This progression showcases how normative pressures, such as adhering to professional and legal standards, shape the evolution of ventures in their quest to align with institutional expectations while maintaining their original mission.

These mechanisms collectively illustrate how not-for-profit GovTechs adapt to align with institutional expectations, ensuring trust and legitimacy in their collaborations with public administrations. However, these adaptations may also constrain innovation and grassroots approaches as ventures prioritize professionalization and alignment with external demands.

The findings depict tangible mechanisms of isomorphic change in bridging the gap between CivicTech and GovTech, with GovTech emerging as a growing field, that may enable smart

cities through diffusion of innovation into the public sector [Bh22]. As these ventures gain traction, their ability to balance professionalization with mission-driven objectives becomes critical for their long-term impact and sustainability.

The effects as shown in interaction with government are strengthened further through the professionalization of the field of GovTech as a whole, as visible in Figure 3. Whereas CivicTech has been a staple, but not very much growing field with a few hundred searches per year, GovTech has seen a sharp rise in the last few years, with over a thousand searches in 2023. This goes hand in hand with the expected higher trust that's generated by a more trustworthy field as a whole [Ni10].

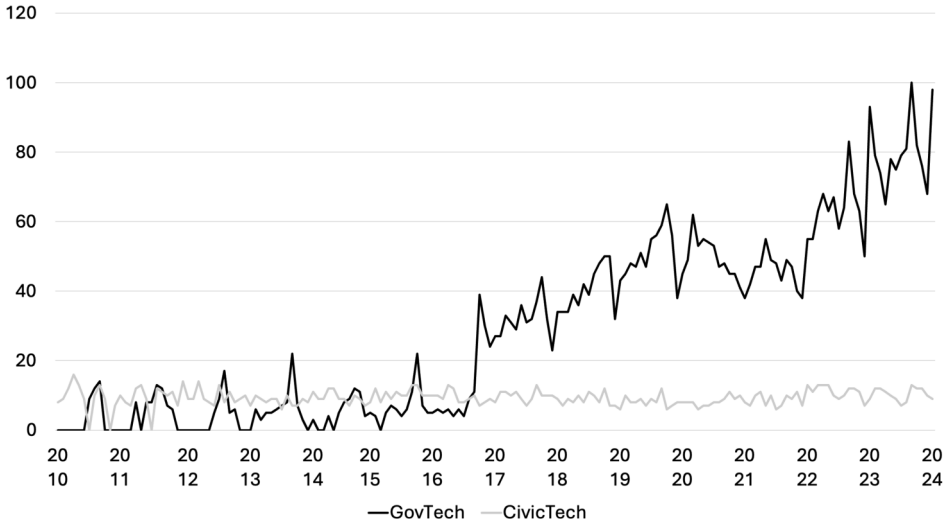


Fig. 3: Google Searches Civic-/GovTech

5 Conclusion and Outlook

In public administration, isomorphic pressures can directly influence the diffusion of innovation. Bureaucratic organizations, inherently risk-averse, often rely on signals of stability and professionalism when engaging with external partners [DP83]. CivicTech ventures, initially characterized by grassroots and non-commercial approaches, face skepticism from public officials [BJ24]. To overcome this, they may opt to adopt practices of professionalization to gain trust such as formalizing governance structures, adopting private-sector business models, or aligning with established commercial norms [Ni10]. These changes do not necessarily alter the core innovation but enhance its perceived legitimacy, easing bureaucratic adoption.

Isomorphism can complement diffusion of innovation theory by addressing public sector-specific barriers, particularly compatibility and observability [RSQ14]. CivicTech ventures that mirror commercial practices signal reliability, aligning their innovations with public administrators' expectations. This dynamic suggests that commercialization is a strategic response to isomorphic pressures, enabling CivicTech ventures to diffuse their innovations more effectively into public administrations by becoming commercial GovTech startups.

Answering the research question:

Do civic technology ventures commercialize their organization to attain public administrations' trust through mechanisms of isomorphic change?

CivicTech initiatives, self-proclaimed as not-for-profit, can use the label GovTech as opportunity to position themselves as solution providers for public administrations. However, the study has several limitations underscoring the need for further research. The sample size was small, focusing on six not-for-profit ventures introduces potential bias, as founders may emphasize their social motivations. This sample does not capture the full spectrum of GovTech, particularly ventures that are purely for-profit [Bh22]. Interviewees suggested some GovTech ventures were established from the outset as commercial entities, driven by an understanding of mimetic isomorphism in advance and aligning their organizational design with public sector expectations. We can infer civic technology ventures commercialize to attain public administrations' trust through mechanisms of isomorphic change.

Additionally, the study lacks input from public administration officials, whose perspectives on trust and legitimacy would provide valuable counterpoints to the narratives of founders [FSP20]. A longitudinal examination of success over time could further add to knowledge on the long-term impact of commercialization and isomorphic adaptation.

Future research should address these shortcomings by including a broader range of GovTech ventures, especially for-profit ones, and incorporating quantitative methods [BJ24]. Mapping procurement platforms and analyzing tenders and their solution providers could provide a more comprehensive understanding of the GovTech ecosystem [FSP20]. Furthermore, investigating the relationship between isomorphic pressures and the sustained success of GovTech ventures over time would add depth to these findings.

The findings of this study suggest that public administrations, especially targeted smart cities, should recognize the role of commercialization in fostering trust and legitimacy among CivicTech ventures. By understanding the pressures these ventures face and the strategic adaptations they make, smart city managers can design procurement and partnership frameworks that balance their need for professionalism with the flexibility to support grassroots innovation [Ne19; Wi24]. Such an approach could reduce public expenditure from process costs and enhance the ability of GovTech ventures to contribute effectively to public sector modernization through the diffusion of innovative of solutions for citizens value. Accordingly, the isomorphic mechanisms appear as deliberate ways to match public needs on paper, without changing the ventures central identity and offering.

References

[Ac24] Ackermann, L.; Akcabay, S.; Benabbas, A.; Khalil, R. E.; Nicklas, D.: Enhancing Data Quality and Collaboration in Participatory Climate Data Crowdsensing. In: 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE, pp. 655–660, 2024.

[Bh22] Bharosa, N.: The rise of GovTech: Trojan horse or blessing in disguise? A research agenda. Government Information Quarterly 39/, p. 101692, 2022, ISSN: 0740624X.

[BJ24] Bharosa, N.; Janowski, T.: The GovTech Challenge - GovTech and Public Value Creation. In: Proceedings of the 25th Annual International Conference on Digital Government Research. ACM, New York, NY, USA, pp. 1043–1045, 2024.

[CD21] Corbett, E.; Dantec, C. L.: Designing Civic Technology with Trust. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, pp. 1–17, 2021, ISBN: 9781450380966.

[CTS19] Cinar, E.; Trott, P.; Simms, C.: A systematic review of barriers to public sector innovation process. Public Management Review 21/, pp. 264–290, 2019, ISSN: 1471-9037.

[DP83] DiMaggio, P. J.; Powell, W. W.: The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields. American Sociological Review 48/, p. 147, 1983, ISSN: 00031224.

[DT24] Degen, K.; Teubner, T.: Wallet wars or digital public infrastructure? Orchestrating a digital identity data ecosystem from a government perspective. Electronic Markets 34/, p. 50, 2024, ISSN: 1019-6781.

[Eu24] EuropeanCommission: Regulation (EU) 2024903 of the European Parliament and of the Council of 13 March 2024 laying down measures for a high level of public sector interoperability across the Union (Interoperable Europe Act), 2024, URL: https://eur-lex.europa.eu/eli/reg/2024/903/oj.

[Fl20] Floridi, L.: The Fight for Digital Sovereignty: What It Is, and Why It Matters, Especially for the EU. Philosophy & Technology 33/, pp. 369–378, 2020, ISSN: 2210-5433.

[FSP20] Ferraris, A.; Santoro, G.; Pellicelli, A. C.: "Openness" of public governments in smart cities: removing the barriers for innovation and entrepreneurship. International Entrepreneurship and Management Journal 16/, pp. 1259–1280, 2020, ISSN: 1554-7191.

[HT22] Holler, R.; Tarshish, N.: Administrative Burden in Citizen-State Encounters: The Role of Waiting, Communication Breakdowns and Administrative Errors. Social Policy and Society/, pp. 1–18, 2022, ISSN: 1474-7464.

[JR14] Johnson, P.; Robinson, P.: Civic Hackathons: Innovation, Procurement, or Civic Engagement? Review of Policy Research 31/, pp. 349–357, 2014, ISSN: 1541-132X.

[Ka21] Kaiser, R.: Qualitative Experteninterviews. Springer Fachmedien Wiesbaden, 2021, ISBN: 978-3-658-30254-2.

[Ki15] Kitchin, R.: Making sense of smart cities: addressing present shortcomings. Cambridge Journal of Regions, Economy and Society 8/, pp. 131–136, 2015, ISSN: 1752-1378.

[Ki22] Kitchin, R.: Conceptualising smart cities. Urban Research & Practice 15/, pp. 155–159, 2022, ISSN: 1753-5069.

[Kl10] Klein, P. G.; Mahoney, J. T.; McGahan, A. M.; Pitelis, C. N.: Toward a theory of public entrepreneurship. European Management Review 7/, pp. 1–15, 2010, ISSN: 1740-4754.

[KM18] Kattel, R.; Mazzucato, M.: Mission-oriented innovation policy and dynamic capabilities in the public sector. Industrial and Corporate Change 27/, pp. 787–801, 2018, ISSN: 0960-6491.

[KP16] Kirchberger, M. A.; Pohl, L.: Technology commercialization: a literature review of success factors and antecedents across different contexts. The journal of technology transfer 41/, pp. 1077–1112, 2016.

[KS22] Kilian, M.; Seifert, J.: Die Lage von GovTech in Deutschland Eine repräsentative Umfrage unter deutschen GovTech-Startups, 2022.

[LC20] Li, J.; Chung, K.: Decomposing Isomorphism: What Drives Similarity in the Adoption of New Public Management? Administration & Society 52/, pp. 375–404, 2020, ISSN: 0095-3997.

[Ma11] Mazzucato, M.: The entrepreneurial state. Soundings 49/, pp. 131–142, 2011, ISSN: 1362-6620.

[Me22] Mergel, I.; Ulrich, P.; Kuziemski, M.; Martinez, A.: Scoping GovTech dynamics in the EU, 2022.

[Ne19] Neumann, O.; Matt, C.; Hitz-Gamper, B. S.; Schmidthuber, L.; Stürmer, M.: Joining forces for public value creation? Exploring collaborative innovation in smart city initiatives. Government Information Quarterly 36/, p. 101411, 2019, ISSN: 0740624X.

[Ni10] Nicholls, A.: The Legitimacy of Social Entrepreneurship: Reflexive Isomorphism in a Pre–Paradigmatic Field. Entrepreneurship Theory and Practice 34/, pp. 611–633, 2010, ISSN: 1042-2587.

[Ni11] Niehaves, B.: Iceberg ahead: On electronic government research and societal aging. Government Information Quarterly 28/, pp. 310–319, 2011, ISSN: 0740624X.

[NK24] Niehaves, B.; Klassen, G.: GovTech in Europe: influencing factors, common requirements and recommendations, 2024.

[Ph15] Phillips, W.; Lee, H.; Ghobadian, A.; O'Regan, N.; James, P.: Social Innovation and Social Entrepreneurship. Group & Organization Management 40/, pp. 428–461, 2015, ISSN: 1059-6011.

[RSQ14] Rogers, E. M.; Singhal, A.; Quinlan, M. M.: Diffusion of innovations. In. Routledge, pp. 432–448, 2014.

[Sa19] Saldivar, J.; Parra, C.; Alcaraz, M.; Arteta, R.; Cernuzzi, L.: Civic Technology for Social Innovation. Computer Supported Cooperative Work (CSCW) 28/, pp. 169–207, 2019, ISSN: 0925-9724.

[Sc16] Schrock, A. R.: Case Study: Code for America—Scaling Civic Engagement through Open Data and Software Design. In. The MIT Press, pp. 217–220, 2016.

[SKT22] Siokas, G.; Kelaidi, V.; Tsakanikas, A.: The smart city as a hub for nourishing public-private partnerships. Sustainable Cities and Society 76/, p. 103466, 2022, ISSN: 22106707.

[SVB09] Sud, M.; VanSandt, C. V.; Baugous, A. M.: Social Entrepreneurship: The Role of Institutions. Journal of Business Ethics 85/, pp. 201–216, 2009, ISSN: 0167-4544.

[To13] Townsend, A. M.: Smart cities: Big data, civic hackers, and the quest for a new utopia. WW Norton & Company, 2013.

[VBB13] Venkatesh, V.; Brown, S. A.; Bala, H.: Bridging the Qualitative-Quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Systems. MIS Quarterly 37/, pp. 21–54, 2013, ISSN: 02767783.

[Wi24] Wimmer, M. A.: Die Intelligenz der Städte. Gesellschaft für Informatik, 2024.

[YT21] Yoshida, M.; Thammetar, T.: Education Between GovTech and Civic Tech. International Journal of Emerging Technologies in Learning (iJET) 16/, p. 52, 2021, ISSN: 1863-0383.

[Zh17] Zhang, K.; Ni, J.; Yang, K.; Liang, X.; Ren, J.; Shen, X. S.: Security and Privacy in Smart City Applications: Challenges and Solutions. IEEE Communications Magazine 55/, pp. 122–129, 2017, ISSN: 0163-6804.

[ZN16] Zhou, L.; Nunes, M. B.: Formulating a Framework for Desktop Research in Chinese Information Systems. In. Pp. 307–325, 2016.

Interact360: A new Paradigm for Interacting with a Future Smart City

Lars Mathuseck [1], Sebastian Lange [2], and Klaus David [3]

Abstract: As more and more cities worldwide move towards a smart city, the information available to residents grows. Consequently, novel approaches are required to interact with a smart city. In this work, we demonstrate the *Interact360* framework, which is the foundation for mobile platforms to interact with a smart city through gestures and Augmented Reality (AR). In our vision, the user of such Apps can easily discover artifacts around the current location, based on context-aware filtering. Artifacts are located around the user, and thus a circular discover-view is proposed to visualize such artifacts. The user can then turn his device in the direction of different artifacts. Once the user turns his device in the direction of an artifact, the user receives haptic and acoustic feedback, using the speaker and the vibration motors of the Smartphone. After selecting an artifact, the user can raise the smartphone as though they are about to capture an image with the camera. Based on the artifact, different information is visualized in Augmented Reality with spatial relation to the artefacts. For example, in the case of a bus stop, the timetable is presented to the user. In addition, the user can perform context-based actions, such as purchasing a ticket. To accomplish this, the user can either swipe and tap on the display or employ basic gestures, such as gently rotating the device. These gestures are detected using the Inertial Measurement Unit (IMU) sensor of the smartphone in combination with on-device machine learning. To demonstrate feasibility, we used the *Interact360* framework to create two applications, specifically targeting Android and iOS. To summarize, the *Interact360* framework facilitates the development of innovative human interaction interfaces designed for context-aware and user-friendly applications in Smart city environments.

Keywords: Augmented Reality, Smart City, HMI, Gestures

1 Introduction

More and more cities around the world head in the direction of smart cities, for example, Singapore [SS24], Barcelona [BA13], and others. Smart cities bring many benefits to their citizens and visitors, such as:

[1] University of Kassel, Chair for Communication Technology – ComTec, Wilhelmshöher Allee 73, Kassel, 34121, lars.mathuseck@comtec.eecs.uni-kassel.de, https://orcid.org/0000-0001-8945-5180
[2] University of Kassel, Chair for Communication Technology – ComTec, Wilhelmshöher Allee 73, Kassel, 34121, sebastian.lange@comtec.eecs.uni-kassel.de, https://orcid.org/0009-0009-1959-1000
[3] University of Kassel, Chair for Communication Technology – ComTec, Wilhelmshöher Allee 73, Kassel, 34121, david@uni-kassel.de, https://orcid.org/0000-0002-6600-9329

doi:10.18420/BTW2025-102

1. **Improved Urban Mobility**: Technologies like IoT and AI can optimize traffic flow, reduce congestion, and promote efficient public transportation and making commuting easier and faster.
2. **Enhanced Sustainability**: Energy-efficient systems, waste management, and renewable energy integration, reducing environmental impact and promoting green living.
3. **Increased Public Safety**: Real-time monitoring systems, connected devices, and data-driven decision-making enhance emergency response times and overall safety for residents.

One significant challenge in the development of smart cities lies in facilitating user interaction with digital artifacts. Digital artefacts can comprise traffic lights, IOT Sensors, Bus/tram/underground information displays, and many more but also "analog" artefacts like an old church, monuments, and even a famous rock. Such analog artefacts become digital ones, if we have a digital representation, e.g. a website with information about this old church, combined with its geolocation.

We would like to address the following challenge:
How to best find and interact with the ever-increasing digital artefacts of future smart cities?

A commonly adopted approach involves leveraging mobile devices that users already possess, enabling interaction through dedicated mobile applications. Currently, numerous such applications exist, often with varied user interfaces and intricate in-app experiences. Additionally, different use-cases are usually spitted into multiple Apps, making it unclear which Apps to use. This diversity can pose challenges, particularly for users with limited technological expertise.

The increasing number of digital artifacts presents a significant challenge in managing and accessing relevant information effectively. To address this issue, context-aware filtering mechanisms are essential for delivering information that cover the users' current needs. Such mechanisms must be designed to accommodate both novice and expert users, ensuring an intuitive and efficient user experience. Conventional solutions, such as map-based applications, often lack efficiency and fail to establish a natural interaction between the Smartphone and the smart city.
Consequently, in this work, we propose a novel approach to interact with a smart city.

This interaction is based on gestures and Augmented Reality (AR). We envision a smart city interaction solution, in which the user can easily discover and interact with digital artifacts around the current location of the user. The artifacts and possible interactions are filtered in a context-aware manner, ensuring that the user interface presents only the actions relevant to the user's needs.

The remainder of this paper is structured as follows.
Section 2 shows out proof-of-concept iOS App for smart city interaction. Section 3 provides an overview of the framework we have developed for Augmented Reality Apps. In section 4, we introduce the gestures we propose to interact with the smart city. In section 5 we show an evaluation of user tests of the App. Finally, this paper is concluded in section 6.

2 Interacting with a Smart City

To demonstrate the feasibility of our vision, we created two applications, specifically targeting Android and iOS, as a proof-of-concept.

Fig. 1: Use-case: Public transport schedule information

When started, the user of the App sees a map with a circle around his current location. Available digital artifacts, located around the user, are visualized in this circular discover-view. The user can then turn his device in any direction, to lock-in on one specific artifact. When locked to an artifact, the user receives haptic, acoustic, as well as visual feedback, using the vibration motors of the Smartphone, the speaker and its screen, respectively. This allows the user to feel, hear and see a nearby artifact we wanted to discover, resulting in completely new and very intuitive user experience. After selecting an artifact, the user can raise the smartphone as though they are about to capture an image with the camera. Based on the artifact, different information or possibilities of interaction are visualized in Augmented Reality with spatial relation to the artifacts, as shown in Fig. 1 (right).

In this example use-case, the user selected a tram stop. Therefore, the user receives information about the artifact, which in this case is timetable information of the specific tram stop, and its distance here 28 meters. Based on the artifact, the user may perform context-aware actions, such as buying a ticket for public transport. To accomplish further actions, the user of the App can either swipe and tap on the display, as they are accustomed to, or perform basic gestures, such as gently rotating the device in one direction. Such gestures allow to select different actions, such as scrolling the timetable in the future or buying a ticket.

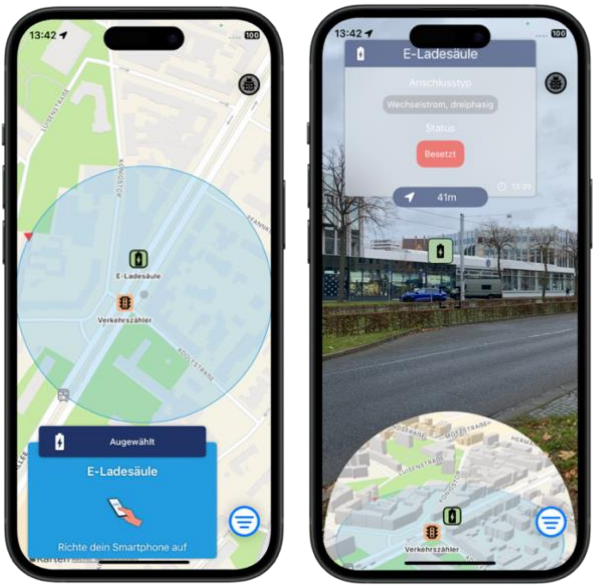


Fig. 2: Use-case: Charging point occupation information

Another use-case is demonstrated in Fig. 2. In this case, the user locked-in on a charging station for electric cars. When performing the gesture of raising his smartphone, the spatial location of the charging station is shown to the user, together with information about the charging point. This information is obtained by an Application Programming Interface (API) provided by the smart city. Context-aware actions, such as short-term reservations of the charging point, or receiving information about other available charging points could be implemented when available. Another use case involves receiving parking information. By pointing a smartphone at a parking garage entrance, users can instantly view available spaces and take actions such as paying for a parking ticket via gestures or screen taps.

Fig. 3: Interacting with a cluster of artefacts

As the number of artifacts in a smart city continues to grow, the clustering of artifacts within a specific direction may become necessary. Fig. 3 illustrates such a scenario, where multiple artifacts are clustered. The user is presented with a stacked arrangement of these artifacts, enabling interaction through swipe gestures on the touchscreen or a gesture.

These examples demonstrate the potential advantages of a unified, user-centric smart city application capable of supporting a diverse range of use cases. By integrating various functionalities into a single, cohesive application, this approach reduces the complexity associated with navigating multiple disparate applications. Compared to existing smart city applications, which often present fragmented and inconsistent user experiences, this vision offers improved usability and interaction efficiency. Such a streamlined design enhances accessibility, promotes inclusivity, and facilitates broader adoption, thereby advancing the overall effectiveness of smart city solutions. However, existing smart city applications and infrastructure can be combined with this App to gain the seamless user experience.

3 Interact360 Framework

In this section, we present the Interact360 framework, which was developed as common base for the envisioned smart city application, allowing rapid development on top of this framework.

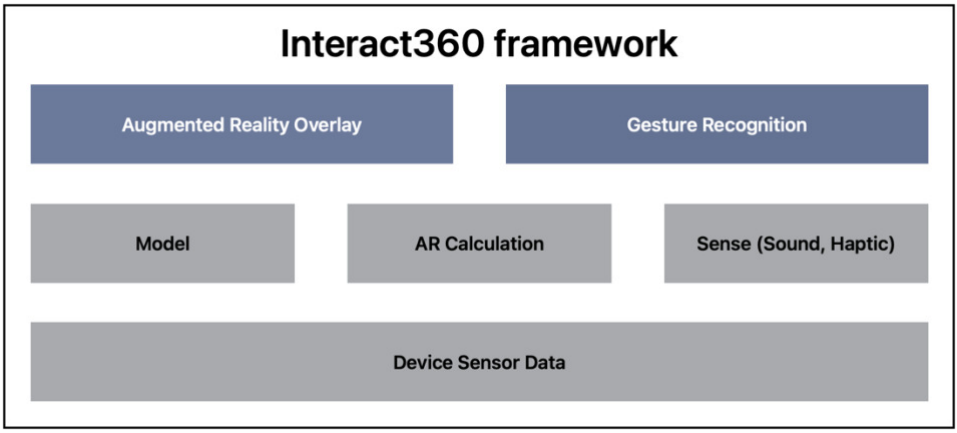


Fig. 5: Overview of the components of the Ineract360 framework

The Interact360 framework comprises two primary components: Augmented Reality Overlay and Gesture Recognition. The Augmented Reality Overlay is fully equipped for integration into mobile applications and provided an out-of-the-box API for mobile developers to display artifacts in AR. Based on the mobile platform, the native UI framework is supported. For iOS, this is the Swift UI framework. For Android, the developers can work directly with Jetpack Compose. The Augmented Reality overlay is used for visualization, based on the calculation of the "AR Calculation"-module.

The *AR Calculation module* is responsible to transform geo coordinates in the WGS84 format with additional altitude information to the screen coordinate system. This is done by first transforming WGS84 coordinated to Earth-centered Earth-fixed (ECEF) coordinates.
These coordinates are then transformed to East-North-Up (ENU) coordinates, based on the user's current position and altitude, resulting in a vector pointing in the direction of the WGS84 coordinates.
Finally, a camera projection matrix can be used to convert the three-dimensional vector (ENU coordinates) to two-dimensional pixel coordinates, by using the current rotation matrix of the smartphone. [Py16], [Ku07]

The "Gesture Recognition"-module is another important module of the Interact360 framework, responsible to detect and differentiate gestures performed by the user.
A gesture is a certain movement of the device, which can be recognized by the Inertial Measurement Unit (IMU) integrated in smartphones, in combination with machine learning. The Interact360 framework comes with an integrated machine learning model for on-device sensor data analysis and machine learning based classification. This allows the user to perform easy to remember gestures to interact with the real-world.

The “Sense” module of the Interact360 framework is responsible to provide haptic and acoustic feedback to the user. This module uses smartphone sensor data, to determine if a user points his device towards an artefact and is therefore in the process of locking-in. This module is crucial in helping to provide an intuitive user experience, where the user can feel and hear the interaction with the smart city.

4 Gestures

As described in Section 2, the user can either interact with the smart city using his Smartphone by tapping on the screen, or, by performing basic gestures. In our vision, different gestures are implemented for different actions. These gestures are shown in Fig. 6.

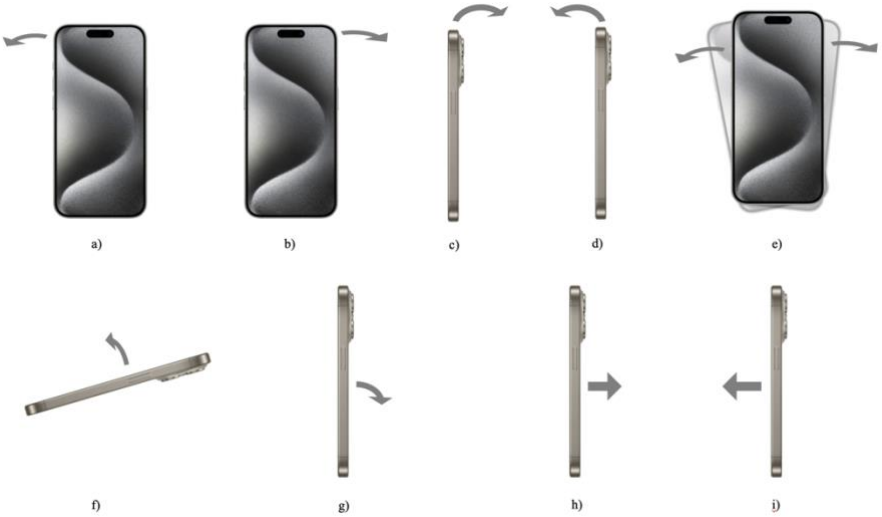


Fig. 6: Gestures the user can perform to interact with the smart city.

The gestures are:

- **Tilt to left:** The user tilts his device to the left, while the device is in a portrait position. This gesture can be used to vertical scroll (left).
- **Tilt to the right:** The user tilts his device to the right, while the device is in portrait position. This gesture can be used to vertical scroll (right).
- **Tilting backwards:** The user tilts his device backwards, while the device is in portrait position. This allows the user to scroll horizontal or, in a special context, further away.

- **Tilting forwards:** The user tilts his device forwards, while the device is in portrait position. This allows the user to scroll horizontal or, in a special context, closer towards the current position.
- **Shaking the device**: The user carefully shakes his device to undo an action he previously performed.
- **Lifting up**: The user lifts up his device from a natural holding position. This action is used to activate the AR-mode.
- **Lifting down**: The user lowers their device back to a natural holding position. This action is used to deactivate the AR-mode.
- **Pushing**: The user pushes the device away, as if he would push a button. This action is used to perform a virtual button press.
- **Pulling**: The user pulls the device towards himself. This action is used to enlarge content on the screen / bring something *closer* to the user.

To recognise such gestures, we use sensor data from the in-built Smartphone IMU sensor. This data is segmented using a sliding window and then classified using on-device machine learning. Although many publications have already shown how to detect such gestures, these publications focus on gesture detection while standing still. [LC12], [MP13], [MP14], [Wa12], [ZLL14], [Ya18]
In contrast, our approach should also work, if the user is currently walking, which required us to build a custom classifier to cover this use-case.

5 User Tests

For better understanding of the interaction between the smart city and the user, we did first user tests. Citizens had the opportunity to test the App and provide feedback through a survey. The survey covered demographic data, participants' experience with smartphones and augmented reality (AR), and their evaluation of the App's usability and user-friendliness. Participants were also asked for improvement suggestions and the types of information they would like to explore within the App.

Five people participated in the test: two under the age of 20 and three over 50. Three were female, and two were male. All participants had extensive smartphone experience, though three used their devices rarely or regularly, while two used them very frequently. Three of the five participants had no prior experience with AR.

The interaction between the map and AR-View was mostly rated positively. The touchless interaction concept stood out, receiving "good" to "very good" ratings from most participants. One concern was that an increasing number of artifacts, such as restaurants, could clutter the AR view. A participant highlighted the importance of a clear introduction or tutorial to help new users understand the App's features.

Additional suggestions included community-driven content to allow users to add their own data and a navigation feature.

Participants expressed a preference for practical, everyday information. Departure times for public transportation, parking spots, and attractions were rated as particularly relevant by three participants. Two found restaurant information useful, while one showed interest in smart city sensor data, such as environmental metrics or traffic counts. For sensor data, it was suggested to include insights into smart city processes, such as how intelligent trash bins work and their impact on urban cleanliness.

The users found the App innovative and intuitive. The tests provided valuable feedback to refine the concept further.

6 Conclusion and Future Work

In this work, we have presented the Interact360 framework, a pioneering solution for interacting with the smart cities of the future. The framework enables the development of mobile applications for both Android and iOS platforms, introducing a novel paradigm for intuitive user interaction with smart city environments. By leveraging Augmented Reality (AR) and on-device machine learning, users can perform basic gestures to interact seamlessly with digital artifacts.
These gestures, detected using the smartphone's Inertial Measurement Unit (IMU) and advanced machine learning techniques, are combined with haptic, acoustic, and visual feedback to create an immersive and engaging user experience. The spatial visualization of data in AR further enhances this interaction, providing a direct relationship between the information and its real-world context.
Designed to cater to both novice and expert users, the Interact360 framework delivers a context-aware, user-friendly interface that redefines how we explore and interact with digital artifacts in smart cities. This innovation not only simplifies the user experience but also paves the way for more natural and accessible smart city applications.
Most of the Interact360 framework is already available as open-source code for the iOS[1] and Android[2] platform.
In the future, we plan to extend the user tests and deploy the smart city Apps to multiple cities.

[1]https://software.opencode.de/project/2905
[2]https://software.opencode.de/project/2906

Bibliography

[BA13] Bakıcı, Tuba ; Almirall, Esteve ; Wareham, Jonathan: A Smart City Initiative: the Case of Barcelona. In: *Journal of the Knowledge Economy* Bd. 4 (2013), Nr. 2, S. 135–148.

[SS24] Sipahi, Esra Banu ; Saayi, Zabihullah: The world's first "Smart Nation" vision: the case of Singapore. In: *Smart Cities and Regional Development (SCRD) Journal* Bd. 8 (2024), Nr. 1, S. 41–58.

[LC12] Lee, Myeong-Chun ; Cho, Sung-Bae: Mobile Gesture Recognition using Hierarchical Recurrent Neural Network with Bidirectional Long Short-Term Memory. In: *The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Barcelona, Spain, 2012.

[MP13] Marasovic, Tea ; Papic, Vladan: A novel feature descriptor for gesture classification using smartphone accelerometers. In: *2013 IEEE Symposium on Computers and Communications (ISCC)*. Split, Croatia : IEEE, 2013.

[MP14] Marasovic, Tea ; Papic, Vladan: User-dependent gesture recognition on android handheld devices. In: *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. Split, Croatia : IEEE, 2014.

[Wa12] Wang, Xian ; Tarrío, Paula ; Metola, Eduardo ; Bernardos, Ana M. ; Casar, José R.: Gesture Recognition Using Mobile Phone's Inertial Sensors. In: Omatu, S. ; De Paz Santana, J. F. ; González, S. R. ; Molina, J. M. ; Bernardos, A. M. ; Rodríguez, J. M. C. (Hrsg.): *Distributed Computing and Artificial Intelligence*. Berlin, Heidelberg : Springer, 2012.

[Ya18] Yang, Sehoon ; Lee, Sangjoon ; Byun, Yungcheol: Gesture Recognition for Home Automation Using Transfer Learning. In: *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*. Bangkok : IEEE, 2018.

[ZLL14] Zhang, Yi ; Li, Sheng Hui ; Luo, Yuan: Segment Matching Gesture Recognition Algorithm and its Application in Smartphone. In: *Applied Mechanics and Materials* Bd. 511–512 (2014), S. 936–940.

[Py16] Pryss, Rüdiger ; Geiger, Philip ; Schickler, Marc ; Schobel, Johannes ; Reichert, Manfred: Advanced Algorithms for Location-Based Smart Mobile Augmented Reality Applications. In: *Procedia Computer Science* Bd. 94 (2016), S. 97–104.

[Ku07] Kuipers, Jack B.: *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. 5. print., 1. paperback ed., [Nachdr.]. Princeton, NJ : Princeton University Press, 2007. — ISBN 978-0-691-10298-6

(Last revised Dezember 2024)

Customer Centricity in the Smart City

Aspects of an participative approach with service orientation

Christian Schachtner [1]

Abstract: This research examines the application and impact of customer centricity concepts in the context of smart cities. The aim is to analyze how customer-oriented approaches influence the development and implementation of smart city solutions and what potentials arise from this for an improved quality of life and sustainability in urban areas. The success of customer centricity initiatives in cities and municipalities depends to a large extent on the extent to which they succeed in adapting organizational structures, integrating technologies in a meaningful way and always putting people at the center. The study focuses on the integration of citizen needs into technology-supported urban development concepts and explores ways in which smart cities can be made more efficient and citizen-friendly through customer centricity.

Keywords: Smart City, Customer Centricity, Citizen Centricity, Implementation Cases, Data-driven Governance

1 Introduction

Increasing urbanization and digitization pose new challenges for cities. The concept of the smart city promises to improve the quality of life of citizens and make urban processes more efficient through the use of innovative technologies. At the same time, the customer or citizen is increasingly becoming the focus of entrepreneurial and municipal action. Smart cities use innovative technologies and data to improve urban services, improve citizens' quality of life, and promote sustainable development [ABD15].

In terms of smart urban design, regional development and city governance, this development is summarized under the term customer centricity. It analyzes how the transfer of customer-centric approaches to urban development and management can contribute to improved citizen participation and more effective implementation of smart city initiatives.

[1] RheinMain University of applied Sciences, Department Design, Computer Science, Media (DCSM), Unter den Eichen 5, Wiesbaden, 65195, christian.schachtner@hs-rm.de, https://orcid.org/0000-0001-5332-6280

doi:10.18420/BTW2025-103

2 Theoretical basics

Customer centricity describes an approach in which the needs and wishes of customers are placed at the center of all activities with the aim of increasing customer loyalty in the long term [KA18]. In contrast to classic product orientation, the desired customer experience is described in detail. In the context of the smart city, this means that citizens become not only users but also co-designers of urban solutions.

Customer centricity is based on three central conceptual building blocks:

a) Customer Relationship Management (CRM)

b) Customer Journey Management (CJM)

c) Customer Experience Management (CEM)

These building blocks enable companies to systematically manage customer relationships, optimize customer experiences along the entire customer journey, and create positive customer experiences. Customers are not only seen as recipients of services, but are actively involved in value creation processes. The aim is to create value together through the active involvement of citizens in marketing and innovation processes [G22].

The personalization of products, services and communication plays a central role in customer centricity. Customers increasingly expect individualized offers and approaches. Companies must therefore find ways to implement hyper-personalization without neglecting data protection [K23].

Smart cities are urban spaces that aim to make more efficient use of resources, improve quality of life and sustainable development through the use of information and communication technologies (ICT) and data-driven solutions [CDN11]. Key aspects of smart cities include:

- Intelligent infrastructures and networking
- Data-driven decision-making
- Citizen participation and co-creation
- Sustainability and resource efficiency

The idea of the customer-centric smart city goes beyond pure technology implementation. It critically examines why certain technologies are needed, what problems they are supposed to solve, and what power dynamics influence their implementation. Transferring this approach to smart cities requires a rethink in urban planning and management. Instead of isolated technological solutions, the focus is on the holistic optimization of the "citizen experience". Integrated urban development concepts (INSEK) can act as links between integration and urban development policy by

addressing topics and fields of action of municipal integration policy [B22].

Technology is understood as a service that is intended to help citizens and improve their quality of life. The development of customer-centric smart cities requires both continuous and disruptive innovation. Continuous innovations aim to gradually improve existing systems, for example by renovating buildings to make them more energy-efficient or optimising public transport. Disruptive innovations, on the other hand, lead to fundamental changes in urban structures. One example of this is virtual power plants, in which private photovoltaic systems and battery storage systems are connected to form a decentralized energy grid.

In the development of a conceptual basis for a customer-centric SmartCity movement, the following approaches should be considered in extension of the methodological considerations according to [JVVH19]:

1. Technology-enabled citizen participation

   The study shows that the use of digital platforms and applications can significantly increase citizen participation in urban development processes. Digital participation tools enable citizens to contribute their ideas and concerns directly to planning processes. One example of this is the development of interactive city maps on which citizens can mark suggestions for improvements to their living environment.

2. Data-driven decision-making

   Smart cities use big data and artificial intelligence to optimize city management decisions. The analysis of traffic flows, energy consumption and usage patterns of public spaces provides valuable insights for needs-based urban planning. It is crucial to protect the privacy of citizens and ensure data security.

3. Personalized services

   By using technology, cities can better tailor their services to the individual needs of citizens. Examples include personalized mobility recommendations based on user behavior or automatic notification of relevant administrative processes.

4. Increased efficiency in administration

   The digitization of administrative processes leads to a significant increase in efficiency. Online portals for citizen services, digital document management systems and automated workflows reduce processing times and improve service quality.

The specification with regard to implementation requirements is to be made through the further considerations.

3 Empirical methodology

To investigate the integration of customer centricity into smart city concepts, a mixed-methods approach was chosen that combines quantitative and qualitative research methods. This allows for a comprehensive view of the topic from different perspectives.

This paper is based on a comprehensive literature review and analysis. Scientific articles and reports from the fields of smart cities, customer centricity and urban development were evaluated using the method of science mapping, structured by the design by Pessin [PSY23]. In addition, expert interviews were conducted with urban planners, technology experts and representatives from industry in order to gain practical insights into the application of customer centricity concepts in the smart city context. In particular, the question of practicability is investigated here, whereby case studies with different focus areas in particular are listed in clustered comparisons (see Tab. 1).

Amsterdam	Living Labs and Co-Creation2	- Amsterdam Smart City: A platform that brings together different stakeholders to solve urban challenges - Amsterdam Innovation Arena: A test bed for smart city technologies at the Johan Cruijff ArenA stadium - CityLab: A program that encourages citizens to develop their own solutions to urban problems	- Development of numerous innovative projects in areas such as mobility, energy and circular economy - Strengthening the local innovation ecosystem - High acceptance and use of smart city solutions by citizens
Barcelona	Citizen Engagement and Open Data[3]	- Decidim Barcelona: A digital platform for participatory democracy where citizens can submit proposals and vote on projects - Barcelona Digital City**: A program to promote digital innovation and citizen participation - Open Data BCN: A portal that makes urban data accessible to	- Increased citizen participation in urban decision-making processes - Development of innovative solutions by local startups and citizens - Improved transparency and accountability of the city

[2] For further details see: https://www.smartcitieslibrary.com/smart-city-amsterdam/
[3] For further details see: https://www.smartcitieslibrary.com/smart-city-barcelona-2/

		citizens and businesses	administration
Copenhagen	Citizen Journey Analysis[4]	- Detailed "Citizen Experience Blueprint" that maps citizens' interactions with city services from birth to death - on this basis, potential for improvement is continuously identified and implemented	- Holistic assessment of citizens' needs clarifies divergent interests and needs - AI-powered sentiment analysis tool, evaluates comments and ratings on city services
Helsinki	City as a Service[5]	- Service as Quality approach in which urban services are offered as flexible, personalized services - Mobility as a Service: mobile devices as lead in the use of services including tracking and GPS technology	Central user guidance of services via a mobile solution that can be accessed from anywhere ensures distributed use of services
Singapur	Partnership[6]	"Moments of Life" platform - the "Moments of Life" platform, which offers personalized services to citizens at different stages of life - Comprehensive digital transformation through central software-as-a-service infrastructure	This customer-centric approach has led to a significant increase in citizen satisfaction
Vienna	Services through data analysis[7]	"Digital Agenda Vienna 2025" links data from different urban areas to generate personalized recommendations and services	Citizens automatically receive information about relevant funding or events based on their profile

Tab. 1: Smart City-Cases in the context Customer Centricity

[4] For further details see: https://www.smartcitieslibrary.com/smart-city-copenhagen/
[5] For further details see: https://www.smartcitieslibrary.com/smart-city-helsinki/
[6] For further details see: https://www.smartnation.gov.sg/
[7] For further details see: https://www.smartcitieslibrary.com/smart-city-vienna-3/

Amsterdam relies on a co-creation approach, in which citizens, companies and research institutions work together on smart city solutions, the approach of which offers maximum flexibility and openness in design.

Barcelona is considered a pioneer in the application of customer centricity concepts in the smart city context. The city has developed a comprehensive strategy for citizen participation and data use, the impact of which is remarkable.

In order to capture the needs of citizens holistically, Copenhagen relies on comprehensive citizen journey analyses. On this basis, potential for improvement is continuously identified and implemented.

Helsinki relies on central availability of services via mobile devices. Municipal services are offered as flexible, personalized services. One example is the "Whim" app, which brings together various mobility options in an integrated system, enabling seamless, needs-based mobility.

Singapore is relying heavily on partnerships and citizen involvement in its development into a smart city. Collaborative approaches are intended to create flexible, networked and social urban structures.

The intelligent use of data enables the City of Vienna to generate personalized recommendations and services from different urban areas according to an existing user profile.

In the next step, five semi-structured interviews were conducted with experts from the fields of urban planning, technology and citizen representation. The interviews aimed to gain practical insights into the implementation of smart city projects and to identify potentials and challenges in the integration of customer centricity principles.

The expert interviews were recorded in September 2024 and evaluated with the help of qualitative content analysis. In the process, categories of challenges and solutions were inductively formed and central topics were identified.

4 Conclusions from the study

The following findings can be derived from the cross-sectional consideration of the case studies found and the expert discussions with regard to the application of customer centricity in smart cities. The transfer of customer centricity concepts to urban planning leads to a paradigm shift towards a citizen-centered approach. Instead of top-down decisions, the focus is on the active involvement of citizens.

This can be achieved by:

1. Citizen-centered use of data in urban planning

   - Participatory platforms: Digital platforms allow citizens to contribute ideas, vote on projects and provide feedback [CK19].
   - Co-creation workshops: Citizens are involved in planning processes and develop solutions to urban challenges together with experts.
   - Data-driven needs analysis: By analyzing usage data and citizen feedback, needs can be identified at an early stage and included in planning.
   - Technological interoperability: Different systems and platforms must work together seamlessly.

2. Personalized urban services

Similar to how companies personalize their products and services, cities can tailor their offerings to the individual needs of citizens:

   - Smart mobility: Traffic management systems and mobility apps that suggest optimal routes and means of transport based on real-time data and personal preferences [BDD16].
   - E-government: Digital administrative services that are tailored to the user profile and the life situation of the citizen.
   - Intelligent energy supply: Smart grids and smart meters enable a demand-oriented and efficient energy supply [TA16].

3. Continuous improvement through feedback loops

Customer centricity in smart cities also means continuously gathering feedback and making improvements through various communication channels:

   - Real-time feedback systems: Apps and sensors record user satisfaction and problems in real time.
   - Predictive maintenance: Predictive maintenance of infrastructures based on usage data and citizen feedback.

These aspects continue to follow the framework of agile urban development. This means bundling iterative approaches in urban planning that enable rapid adaptations [H19].

Improved citizen satisfaction and quality of life are geared towards citizens' needs and quality of life through a consistent focus on citizens' needs and quality of life. This can be done through tailor-made solutions in the portfolio of urban services and infrastructures that address the real needs of citizens. Active involvement also promotes

citizens' identification with their city, drives agile city administrations and regions for a stronger sense of community.

This is also ensured by needs-based planning and implementation processes of investments and balancing services in order to avoid inefficiencies in urban processes in a targeted manner through the use of artificial intelligence and predictive analytics. The creation of transparent information on resource consumption in households as neighbourhood recommendations on optimised infrastructures in energy supply or waste management also supported the regional goals for reducing the ecological footprint.

Another aspect is the promotion of innovation through open data and APIs in order to provide and process data-based services with the population. Through Living Labs, cities and municipalities become test beds for innovative solutions that are developed together with citizens and companies. The creation of economic framework conditions for start-up support gives rise to business models according to which innovative start-ups can respond to citizens' needs in urban areas.

4.1 Implementing Customer Centricity

The successful implementation of customer centricity requires profound changes in corporate structures, systems and cultures. In order to establish customer centricity as a guiding principle, the organizational prerequisites must first be created.

These include:

- Adaptation of organizational structures
- Implementation of customer-centric systems and processes
- Development of a customer-oriented corporate culture

These adjustments form the basis for the implementation of the Customer Centricity building blocks. Implementing customer-centric organizational structures often proves to be a challenge. Less than 30% of Fortune 500 organizations have implemented customer-facing structures to date [K23].

Reasons for this are:

- Increased cost and complexity of customer interactions
- Difficult to measure the benefits of customer-centric organizations
- Need for profound changes in established structures

Although technologies such as AI, augmented reality and digital assistants are becoming increasingly important to the customer experience, technology alone is not enough.

Successful customer experience transformation requires the marriage of technological innovation, data-driven insights, and a human-centered approach [S24].

An essential element of customer centricity is the active involvement of customers in marketing and innovation processes. Through co-creation, companies can create value together with customers and optimally adapt products and services to customer needs. In order to increase the acceptance of smart city projects and at the same time meet the need for data protection, many cities are relying on open data and open innovation approaches. By providing open data and involving citizens in innovation processes, more transparency and participation are to be achieved.

However, an excessive focus on data collection also carries risks. The reduction of customers to data points can lead to the actual purpose of customer centricity being lost sight of [KA18].

4.2 Challenges

Despite the great interest in customer centricity in research and practice, many companies fail to implement it consistently. Successfully integrating customer centricity into smart city initiatives requires a comprehensive strategic approach. This should include the following elements [F20]:

- Definition of clear goals and KPIs for the improvement of the citizen experience
- Creation of a detailed citizen journey map for all relevant areas of life
- Establishment of cross-functional teams to optimize customer touchpoints
- Implementation of a data governance framework for responsible data use
- Development of a roadmap for the step-by-step implementation of customer-centric initiatives

A central challenge in the implementation of customer centricity in smart cities is the handling of data. On the one hand, the collection and analysis of customer data enables products and services to be better adapted to individual needs. The challenge for municipalities and companies is to combine and coordinate the various building blocks of a smart city in a meaningful way [D22].

The nationwide introduction of a network of sensors through the integration of IoT or applications based on the 5G and 6G mobile communications standards makes it possible to provide data services over large areas. Possible services such as the use of blockchain technology enable secure citizen participation or identity management for personalized services. With increasing data use and personalization, ethical issues are becoming more important. Companies need to find ways to implement customer centricity without crossing ethical boundaries or abusing customer trust [VRB24].

On the other hand, awareness of data protection and the demand for more privacy is growing worldwide. The equation of data collection and customer centricity, as promoted by digital platforms such as Google and Amazon, is increasingly reaching its limits [G22].

The focus on digital solutions carries the risk of excluding certain population groups. This concerns access to technology. Not all citizens have the necessary devices or skills to use digital services. Age-related barriers could also make it difficult for older generations to adapt to new technologies. Socio-economic factors of the use of certain services could also fail due to financial conditions.

The implementation of customer centricity in smart cities also requires a high level of coordination between different actors. This concerns the cooperation of different actors who have to overcome their silo mentality. Cross-departmental cooperation in the city administration is necessary. Public-private partnerships through the involvement of private companies must also be carefully managed and contractually secured with rights and obligations.

Customer centricity approaches can support the implementation of circular economy concepts by actively involving citizens in recycling and upcycling initiatives. Personalised information and incentives can motivate citizens to behave more sustainably.

4.3 Classification of the findings

The future of customer centricity lies in a holistic approach that optimally combines technological innovations, data-driven insights and human interaction. This is the only way companies can create outstanding customer experiences in the long term and differentiate themselves from the competition. In order to meet these challenges holistically, the following solutions have proven to be promising:

1. Privacy by Design: The integration of data protection measures into the development of smart city solutions from the very beginning.
2. Digital education initiatives: programmes to promote digital skills to enable participation by all citizens.
3. Open data platforms: Making city data available in open formats promotes innovation and transparency.
4. Agile administrative structures: The introduction of agile methods in the city administration allows for more flexible adaptation to citizens' needs.

Future research should focus on examining the long-term impact of smart city initiatives on quality of life and civic engagement. In addition, further studies are needed to find the optimal balance between technological innovation and the preservation of traditional urban qualities.

Future research should focus more on how customer centricity can be implemented in smart cities without neglecting data protection. The question of how urban and rural areas can complement each other in the sense of a holistic smart region also offers interesting starting points for further investigations.

5 Conclution

Customer centricity has established itself as a central concept in corporate management. The consistent alignment of all areas of responsibility with the customer offers great potential for increasing customer loyalty and corporate success. However, companies face a variety of challenges when it comes to implementation. This applies equally to public institutions.

The exemplary case studies such as Barcelona or Amsterdam show that the implementation of customer centricity approaches in smart cities is already leading to tangible results today. At the same time, challenges such as data protection, digital inclusion and the complexity of coordinating different actors must be overcome.

The success of customer centricity initiatives in cities and municipalities depends to a large extent on the extent to which they succeed in adapting organizational structures, integrating technologies in a meaningful way and always putting people at the center. Only by striking the right balance between data use, personalisation and privacy can we ensure the successful performance of the public service in the long term. The development of smart cities with a strong focus on customer centricity is a continuous process that requires constant adaptation and innovation. Cities that successfully follow this path will not only be more efficient and technologically advanced, but above all more livable and citizen-friendly.

Bibliography

[ABD15] Albino, V.; Berardi, U., & Dangelico, R. M.: Smart cities: Definitions, dimensions, performance, and initiatives. Journal of urban technology, 22(1), pp. 3-21, 2015.

[BDD16] Benevolo, C.; Dameri, R. P., & D'auria, B.: Smart mobility in smart city: Action taxonomy, ICT intensity and public benefits. In: Empowering organizations: Enabling platforms and artefacts, Springer International Publishing, pp. 13-28, 2016.

[B22] BMWSB: Migration, Integration and Participation in Integrated Concepts of Urban Development – Guidelines for Municipal Practice published. https://www.staedtebaufoerderung.info/SharedDocs/kurzmeldungen/DE/TopMeldungen/221130_Publikation_INSEK_Migration_Integration_Teilhabe.html, accessed: 24/11/2024.

[CDN11] Caragliu, A.; Del Bo, C., & Nijkamp, P.: Smart cities in Europe. Journal of urban technology, 18(2), pp. 65-82, 2011.

[CK19] Cardullo, P., & Kitchin, R.: Being a ‘citizen’in the smart city: Up and down the scaffold of smart citizen participation in Dublin, Ireland. GeoJournal, 84(1), pp. 1-13, 2019.

[D22] DIN e.V.: Technologie und Mensch in der Kommune von morgen. Impulspapier III zu Normen und Standards – Smart City Smart City Standards Forum. https://www.din.de/resource/blob/916688/080f05531c7a49caac20c569e3d05505/smart-cities-impulspapier-iii-2023-web-data.pdf, accessed: 24/11/2024.

[F20] Fader, P.: Customer centricity: Focus on the right customers for strategic advantage. University of Pennsylvania Press, 2020.

[G22] Gotsch, M. L.: Customer Centricity & Datenschutz - Die Geschichte eines Missverständnisses. Marketing Review St.Gallen, 39(2), pp. 12-19, 2022.

[H19] Haller, S.: Smart Cities and Regions: Die digitale Transformation in der Stadtentwicklung und E-Government. Handbuch E-Government: Technikinduzierte Verwaltungsentwicklung, pp. 519-544, 2019.

[JVVH19] Jordan, Y.; von Viebahn, C., & Haller, S.: Konzeptionierung von Use Cases für die urbane Logistik in einer Schweizer Smart City mittels Design Thinking. Nachhaltige Unternehmensführung: Herausforderungen und Beispiele aus der Praxis, pp. 391-420, 2019.

[KA18] Kalka, R.; Abel, K.: Customer Centricity: Konzeptionelle Grundlagen und Implementierung. Düsseldorf Working Papers in Applied Management and Economics, No. 44, Hochschule Düsseldorf, Fachbereich Wirtschaftswissenschaften, Düsseldorf, 2018.

[K23] KPMG: Die Customer Experience Excellence Studie 2023. https://klardenker.kpmg.de/customer-insights-hub/die-customer-experience-excellence-studie-2023/, accessed: 24/11/2024.

[PSY23] Pessin, V.; Santos, C.; Yamane, L.; Siman, R.; Baldam, R. & Lacerda Júnior, V. A method of Mapping Process for scientific production using the Smart Bibliometrics. MethodsX, 11, 102367, 2023.

[S24] Schachtner, C. (2024). Bürgernahe Umsetzung der offenen Demokratie durch Digitalisierung in der öffentlichen Verwaltung. In: O'Gorman, S., Schuster, G. (eds) Customer Centricity. Springer Gabler, Wiesbaden.

[TA16] Tuballa, M. L., & Abundo, M. L.: A review of the development of Smart Grid technologies. Renewable and Sustainable Energy Reviews, 59, pp. 710-725, 2016.

[VRB24] Velasco, C., Reinoso-Carvalho, F., Barbosa Escobar, F., Gustafsson, A., & Petit, O.: Paradoxes, challenges, and opportunities in the context of ethical customer experience management. Psychology & Marketing, 41(10), pp. 2506-2524, 2024.

Verwendung von MobiTopp als Grundlage für die Generierung von Bewegungsverläufen mit SUMO

Lukas Kneis [1]

Abstract: Aufgrund datenschutzrechtlicher Bestimmungen ist die Herausgabe von Datensätzen mit genauen Bewegungsverläufen in der EU mit hohen Hürden verbunden. Falls für Forschungszwecke ein Datensatz mit realistischem Bewegungsverhalten benötigt wird, wird somit häufig aus Datensätzen aus dem außereuropäischen Ausland zurückgegriffen. Diese Daten spiegeln allerdings nicht zwingend das Mobilitätsverhalten in Europa oder Deutschland wider, was realistische Vorhersagen im Smart City Kontext erschwert. Dieser Demonstrator soll zeigen, dass die frei verfügbare Simulation der Stadt Rastatt genutzt werden kann, um realistische Bewegungsverläufe zu erhalten. Hierfür wird die in MobiTopp generierte Verkehrsnachfrage als Grundlage für die agentenbasierte Verkehrssimulation SUMO verwendet. Diese erlaubt die Generierung detaillierter Trajektorien bspw. für die Vorhersage des Verkehrsverhaltens von Radfahrern.
Als Basis für den Demonstrator dient eine MobiTopp Simulation der Stadt Rastatt, perspektivisch ist allerdings auch die Verwendung anderer MobiTopp Simulationen denkbar. Basierend auf Befragungen wurden demografische Daten ermittelt, welche die Grundlage für diese Simulation darstellt. Bei MobiTopp handelt es sich um eine agentenbasierte Simulation der Verkehrsnachfrage. MobiTopp verfügt allerdings über keine konkreten Routen für die einzelnen Agenten sondern verwendet lediglich sogenannte Traffic Analysis Zones. Um konkrete Bewegungsverläufe zu erhalten, ist also eine Verknüpfung mit einer Verkehrssimulation nötig. Genutzt wird in diesem Fall die mikroskopische Verkehrssimulation SUMO. Der Demonstrator verknüpft die beiden Simulationen in Form einer Toolchain und generiert so GPS Traces, aber auch andere von SUMO unterstützte Formate.
MobiTopp steht am Anfang der Toolchain und generiert zunächst eine synthetische Population bestehend aus einzelnen Haushalten. Diese Haushalte werden anschließend mit Agenten befüllt, welche im Verlauf einer Woche basierend auf Eigenschaften wie Alter und Berufstätigkeit eine Verkehrsnachfrage erhalten. Diese Nachfrage verfügt neben Start- und Zielposition auch über einen gewählten Verkehrsmodus und einen Startzeitpunkt. Diese Verkehrsnachfrage dient als Grundlage für die Erstellung einer SUMO Simulation.
Um eine SUMO Simulation ausführen zu können, wird ein Straßennetz benötigt. Anhand der Start- und Zielpositionen in der Verkehrsnachfrage wird ein geeigneter Ausschnitt gewählt und ein Import aus OpenStreetMap durchgeführt. Optional kann der Kartenbereich auch verkleinert werden, um den Speicherbedarf zu reduzieren. Auf der so importierten Karte können anschließend Trips durchgeführt werden, welche sich aus den Daten aus MobiTopp ergeben. SUMO ermöglicht hierbei ein iteratives Routing, wodurch sich die Verkehrslast über das Straßennetz verteilt. Ebenfalls möglich ist ein Import von GTFS Daten für die Einbindung des ÖPNV. Um die Laufzeit zu verringern, ist eine Einschränkung des Simulationszeitraums möglich, bspw. ein Zeitraum von Stunden statt der vollen Woche. Abschließend kann SUMO die Daten in einem letzten Simulationsdurchlauf abspeichern.
Dieser Demonstrator ist im Rahmen des Forschungsprojekts „ANYMOS - Kompetenzcluster Anonymisierung für vernetzte Mobilitätssysteme“ entstanden. Danke an das Bundesministerium für Bildung und Forschung BMBF für die Finanzierung im Rahmen von NextGenerationEU der Europäischen Union (Förderkennzeichen 16KISA086).

Keywords: Bewegungsdaten, Verkehrssimulation, MobiTopp, SUMO

[1] FZI Forschungszentrum Informatik, Software Engineering, Haid-und-Neu-Str. 10–14, 76131 Karlsruhe, Deutschland, kneis@fzi.de, https://orcid.org/0009-0009-4778-4969

doi:10.18420/BTW2025-104

SensBee - A Sensor Data Backend for Smart City Applications

Timo Räth [1], Laurin Martins [1], and Kai-Uwe Sattler [1]

Abstract:

The number of Smart City applications is increasing at a steady pace, as the amount of sensor data being generated, transmitted, and processed continues to grow. The ongoing development of new public data-driven use-cases, such as traffic monitoring, parking lot capacity tracking, and visitor counting, offers users the possibility to link with real-world applications in their local communities. Likewise, substantial research has been conducted in recent years examining the opportunities and limitations of modern Smart City solutions [Go24, Si22], as well as the integration of heterogeneous Internet of Things (IoT) and edge devices [MBD24, Kh20]. As a result, novel Smart City backend solutions and frameworks, such as FIWARE [Ci19] or CityPulse [Pu16], are introduced to manage the generated sensor data and provide interfaces for visual and interactive data dashboards to inform users about the current status of the applications.

In this paper, we present *SensBee*[2], a new Rust-based sensor data backend solution, designed for the lightweight management of sensors and sensor users in the context of smart city applications. SensBee follows the concept of an *Open City*, which aims to involve the general public in the deployment, development, and utilization of sensor devices and applications. For this reason, SensBee offers a comprehensive range of flexible options, to register, manage, and access sensors for different users, and user groups. For instance, users may register their own private sensor devices and share the produced data with the general public for further processing or analysis purposes. Likewise, existing public user or system sensor devices can be accessed and integrated into own diverse Smart City applications, such as the tracking of visitors in the local swimming pools. Furthermore, our backend offers flexible data retrieval interfaces that can be integrated directly into established data visualization solutions, such as Grafana dashboards[3], to create interactive presentations of the recorded sensor data. To facilitate the integration of a multitude of heterogeneous sensor devices, our system offers configurable data ingestion interfaces for an extensible range of protocols, including HTTP and MQTT.

During the demonstration, we will showcase the workflow and capabilities of our sensor data backend utilizing real-world examples from our home city, Ilmenau. Visitors may try out the core functionalities of the system, such as registering new sensors or accessing available data on their own, and engage in a discussion regarding potential use-cases for their local Smarty City environments. Furthermore, as the SensBee project is still in an active development phase, visitors may contribute their experiences with previous sensor data backends and Smart City applications, as well as propose potential opportunities for improvements or additional functionalities.

Keywords: Smart City, Sensors, Sensor Data, Open City

[1] TU Ilmenau, Database and Information Systems Group, Ilmenau, Germany,
timo.raeth@tu-ilmenau.de, https://orcid.org/0000-0002-3452-4529;
laurin-hendrik.martins@tu-ilmenau.de, https://orcid.org/0009-0004-7263-1474;
kus@tu-ilmenau.de, https://orcid.org/0000-0003-1608-7721

[2] https://github.com/dbis-ilm/sensbee

[3] https://grafana.com/grafana/dashboards

doi:10.18420/BTW2025-105

Bibliography

[Ci19] Cirillo, Flavio; Solmaz, Gurkan; Berz, Everton; Bauer, Martin; Cheng, Bin; Kovacs, Ernö: A Standard-Based Open Source IoT Platform: FIWARE. IEEE Internet of Things Magazine, 2:12–18, 09 2019.

[Go24] Goumopoulos, Christos: Smart City Middleware: A Survey and a Conceptual Framework. IEEE Access, PP:1–1, 01 2024.

[Kh20] Khan, Latif U.; Yaqoob, Ibrar; Tran, Nguyen; Kazmi, S.M.; Nguyen Dang, Tri; Hong, Choong Seon: Edge-Computing-Enabled Smart Cities: A Comprehensive Survey. IEEE Internet of Things Journal, PP:1–1, 04 2020.

[MBD24] Motta, Rebeca; Batista, Thaís; Delicato, Flavia: The Intersection of the Internet of Things and Smart Cities: A Tertiary Study. Journal of Internet Services and Applications, 15:325–341, 09 2024.

[Pu16] Puiu, Dan; Barnaghi, Payam; Tönjes, Ralf; Kumper, Daniel; Ali, Muhammad Intizar; Mileo, Alessandra; Parreira, Josiane; Fischer, Marten; Kolozali, Şefki; Farajidavar, Nazli; Gao, Feng; Iggena, Thorben; Pham, Thu-Le; Nechifor, Cosmin-Septimiu; Puschmann, Daniel; Fernandes, Joao: CityPulse: Large Scale Data Analytics Framework for Smart Cities. IEEE Access, 4:1086–1108, 01 2016.

[Si22] Singh, Tarana; Solanki, Arun; Sharma, Sanjay; Nayyar, Anand; Paul, Anand: A Decade Review on Smart Cities: Paradigms, Challenges and Opportunities. IEEE Access, 10:2169–3536, 06 2022.

How to Travel the City? – Making Public Transport Travel Times Explorable for Everyone

Lukas Panzer [1] and Fabian Beck [1]

Keywords: Visualization, Virtual Reality, Public Transport

Public transport networks play a crucial role in urban mobility, providing basic transport for people of all ages and societal groups. However, these networks present a challenge in terms of understandability—the maps and plans that describe routes and schedules are often complex and difficult to interpret; mobile transport apps focus on single-trip connections. Furthermore, these representations do not illustrate which destinations can be reached within a given time frame, or what would be alternative modes of transport. This lack of transparency and practicality not only impedes existing users, but can also detract potential future passengers from using public transport.

Here, visualization can be a tool to simplify complex transport data and make them easier to understand for citizens. By visualizing schedules and travel times in clear, accessible, and engaging representations, we can help a broader audience to gain useful information for their daily travels. In addition, visual comparisons of different modes of transport in a city can provide valuable insights and encourage citizens to consider alternative modes.

Approaches to visualize public transportation systems [Ze14], travel times [GKV21], and the comparison of travel times across different transport modes [ST13] exist within the scientific community. However, they are usually presented for desktop settings in a non-engagingly technical way, for example, the representation of travel times in the form of travel time maps [Ge24]. In contrast, to provide easy-to-understand and engaging representations, we want to compare their visualization potential with that of alternative platforms, including virtual reality (VR) [PB24], public displays [KLS13], and physical projection [PC19]. Moreover, storytelling elements can guide users through the data and interaction options.

By discussing the advantages and disadvantages of different visualization approaches and platforms based on specific examples (own prototypes and other sources), we provide new impulses for municipalities and public transport authorities to improve their communication and interaction with citizens. While we use the city of Bamberg and the transport network association operating in this area as an example, the discussion and results are transferable to other cities.

[1] University of Bamberg, Information Visualization, An der Weberei 5, 96047 Bamberg, Germany,
lukas.panzer@uni-bamberg.de, https://orcid.org/0009-0002-6805-0824;
fabian.beck@uni-bamberg.de, https://orcid.org/0000-0003-4042-3043

doi:10.18420/BTW2025-106

Bibliography

[Ge24] Geoapify: CommuteTimeMap. https://commutetimemap.com/map, 2024.

[GKV21] Gracious, Rony; Kumar, B. Anil; Vanajakshi, Lelitha: Characterizing Bus Travel Time using Advanced Data Visualization Techniques. Transportation in Developing Economies, 7(1):1, 2021.

[KLS13] Kühn, Romina; Lemme, Diana; Schlegel, Thomas: An Interaction Concept for Public Displays and Mobile Devices in Public Transport. In (Kurosu, Masaaki, ed.): Human-Computer Interaction. Interaction Modalities and Techniques. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 698–705, 2013.

[PB24] Panzer, Lukas; Beck, Fabian: Towards Presenting Travel Times in a Bus Network as Immersive and Adaptive Data Stories . In (Kucher, Kostiantyn; Diehl, Alexandra; Gillmann, Christina, eds): EuroVis 2024 - Posters. The Eurographics Association, 2024.

[PC19] Priestnall, Gary; Cheverst, Keith: Understanding visitor interaction with a projection augmented relief model display: insights from an in-the-wild study in the English Lake District. Personal and Ubiquitous Computing, 26(3):593–607, 2019.

[ST13] Salonen, Maria; Toivonen, Tuuli: Modelling travel time in urban networks: comparable measures for private car and public transport. Journal of Transport Geography, 31:143–153, 2013.

[Ze14] Zeng, Wei; Fu, Chi-Wing; Arisona, Stefan Müller; Erath, Alexander; Qu, Huamin: Visualizing Mobility of Public Transportation System. IEEE Transactions on Visualization and Computer Graphics, 20(12):1833–1842, 2014.

On-demand Context Aware Bus Route Profiling: A Study on Enhancing Urban Mobility

Bishal Bhandari [1] and Debasree Das [1]

Abstract: The global population surge has made managing public transit increasingly challenging due to inadequate services, infrastructure, delays, and limited availability [BP12]. This leads to passengers preferring private vehicles over public transit, yet public transportation remains a cornerstone of urban mobility, promoting energy conservation, social equity [Ov21], and reduced traffic congestion [BFL15]. Among public transportation modes like rail, buses, and shuttles, bus systems play a vital role, catering to both short city trips and inter-state travel with moderate passenger loads. However, fixed-route bus systems [Fe19] often fail to meet dynamic urban demands influenced by shifting population density, emerging POIs, and irregular commuter patterns, necessitating a more adaptive approach to rerouting, timetables, and stop placement. For instance, bus stops near railway stations, universities, or offices require more buses during peak hours or festivals compared to holidays or off-seasons. Our research aims to address these challenges by generating context-aware bus routes that adjust service frequency and schedules while proposing dynamic stops based on demand. We define the context in terms of a diverse environmental conditions such as population density, proximity to important POIs like railway stations, offices, banks, universities, other shared mobility stations, etc., seasonality, time of the day & adjacent inter-stop distance. To address significant variations in human mobility patterns across geo-terrains, we aim to make the data-driven system location-invariant by incorporating data from diverse locations, ensuring adaptability [Sh21] across domains (e.g., different European cities). First, we want to generate the demand for a bus service via fusing existing bus route systems comprising bus stop location, #bus_lines per bus stops and their frequencies along with the discussed context. To achieve generalization, the model will be bootstrapped on one location and fine-tuned with domain variance from another to adaptively predict demand. The inferred demand will guide recommendations for bus schedules at existing stops and suggest new bus stop locations with suggested routes. In our initial research, we target two cities from different countries, Bamberg, Germany and Brussels, Belgium due to variation in population, area, and other geo-spatial diversity. For the first city Bamberg, we find $\leq 3\%$ bus stops are served by more than 5 bus lines whereas $\approx 70\%$ bus stops are served by only 1 bus line, which further depicts the limited availability of bus services causing dissatisfaction among the citizens around the city. We also correlate #bus_lines per bus stops with the nearby (radius=500m) population density and PoIs clustered using popularity metric and achieved a Spearman Correlation Co-efficient (SCC) of 0.4412 with p-value= $4.44e^{-16}$ and 0.181 with p-value= $2.89e^{-15}$, respectively. This statistically significant SCC shows that population density is one of the important but not a single factor in positioning bus stops whereas low correlation with the PoIs implies demand for bus stop does not exhibit from only populous zones, further validating the need for the other contexts. Initial findings indicate this work-in-progress research shows promising direction in recommending bus schedules and bus stops to improve passenger satisfaction by addressing urban needs. We aim to evaluate our system by collecting the real-world passenger demand as a ground truth signifying need for a bus service and validate against our generated demand. Finally, to receive the feedback of passengers on recommendations (bus time & bus stop position), we aim to conduct an usability study.

Keywords: Bus Route Optimization, Urban Mobility

[1] University of Bamberg, Bamberg, Germany,
bishal.bhandari@stud.uni-bamberg.de, https://orcid.org/0009-0002-1913-7608;
debasree.das@uni-bamberg.de, https://orcid.org/0000-0003-0172-0280

doi:10.18420/BTW2025-107

Bibliography

[BFL15] Beaudoin, Justin; Farzin, Y Hossein; Lawell, C-Y Cynthia Lin: Public transit investment and sustainable transportation: A review of studies of transit's impact on traffic congestion and air quality. Research in Transportation Economics, 52:15–22, 2015.

[BP12] Buehler, Ralph; Pucher, John: Demand for public transport in Germany and the USA: an analysis of rider characteristics. Transport Reviews, 32(5):541–567, 2012.

[Fe19] Fei, Xiqi; Gkountouna, Olga; Pfoser, Dieter; Züfle, Andreas: Spatiotemporal bus route profiling using odometer data. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 369–378, 2019.

[Ov21] Oviedo, Daniel: Making the links between accessibility, social and spatial inequality, and social exclusion: A framework for cities in Latin America. In: Advances in transport policy and planning, volume 8, pp. 135–172. Elsevier, 2021.

[Sh21] Shao, Wei et al.: FADACS: A Few-Shot Adversarial Domain Adaptation Architecture for Context-Aware Parking Availability Sensing. In: IEEE PerCom. pp. 1–10, 2021.

User-Driven Placement of Urban Digital Twin 3D-Models for Augmented Reality

Bernhard Öder [1] and Sophie Jörg [1]

Keywords: Digital Twin, Augmented Reality

This work presents the idea of combining Augmented Reality (AR) with an urban Digital Twin (DT) to visualize virtual historical content in real-world locations. Content added to the DT can be experienced on-site using a novel manual placement method. This approach leverages user guidance for precise alignment, bypassing technical and environmental limitations common in computer vision methods, offering a robust and accessible solution for urban and historical exploration.

One core technological challenge in Augmented Reality (AR) lies in accurately anchoring virtual content to real-world environments. Visual recognition techniques, which rely on identifying features within reference data of the physical space, are commonly employed for this purpose. Despite the advancements in AR and computer vision algorithms over the past decades [Ki18; ZDB08], no universal method exists for precise placement of virtual objects. Technical limitations, environmental variability, and dependency on reference data continue to impede widespread applicability and scalability of these solutions.

To address these limitations, we propose a user-driven manual placement method. The process involves displaying a semi-transparent reference image of the scene, which the user aligns with the live camera feed of their surroundings. These reference images of the real-world scenes can be taken from the digital twin. This process enables remote content creation, allowing reference images for the AR application to be generated without the need for on-site presence. Once alignment is achieved, a 3D model can be accurately positioned in the correct location. In a historical context, this method can be gamified as an engaging search activity, encouraging users to interact with their environment. Additionally, this approach serves as a reliable alternative in scenarios where computer vision techniques fail, making it a versatile backup for positioning tasks.

Participant questionnaires will assess the perceived registration precision, its consistency across environments, and cases where positioning fails, along with the method's engagement, intuitiveness, and impact on interest in the city or history.

Acknowledgements

The project was supported through the Smart City Research Lab at the University of Bamberg, funded by the City of Bamberg in the scope of the federal program Smart Cities made in Germany.

[1] University of Bamberg, Computer Graphics, Germany,
bernhard.oeder@uni-bamberg.de, https://orcid.org/0009-0007-7999-3381;
sophie.joerg@uni-bamberg.de, https://orcid.org/0000-0002-7910-8553

doi:10.18420/BTW2025-108

References

[Ki18] Kim, K.; Billinghurst, M.; Bruder, G.; Duh, H. B.-L.; Welch, G. F.: Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008–2017). IEEE Transactions on Visualization and Computer Graphics 24 (11), pp. 2947–2962, 2018.

[ZDB08] Zhou, F.; Duh, H. B.-L.; Billinghurst, M.: Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. Pp. 193–202, 2008.

Ageing Smart DSS

Spatial Decision-Support-System für die kommunale Planung

Matthias Berg [1], Benjamin Hassenfratz [2]

Abstract: In den kommenden Jahrzehnten wird der demografische Wandel für deutsche Städte und Gemeinden zunehmend Herausforderungen hinsichtlich der kommunalen Planung mit sich bringen. Dies wird insbesondere an den sogenannten Babyboomern deutlich: Die Jahrgänge zwischen 1955 und 1969 zeichnen sich mit mehr als 1,1 Mio. Geburten pro Jahr als besonders geburtenstark aus [SB23] und treten in den kommenden Jahren in den Ruhestand ein. In der Folge ergibt sich nicht nur eine Lücke auf dem Arbeitsmarkt, sondern es entstehen auch Fragen und Unsicherheiten hinsichtlich der aktuellen und zukünftigen Bedürfnisse der Babyboomer was u. a. Wohnformen, (soziale) Infrastrukturen und Versorgungsangebote betrifft [AH25]. Diese Problemlage bildet den Ausgangspunkt des von der Carl-Zeiss-Stiftung geförderten Projekts *Ageing Smart*, das von der RPTU Kaiserslautern, dem DFKI und dem Fraunhofer IESE bearbeitet wird. Ziel ist die Entwicklung eines Decision-Support-Systems (DSS), das die Entscheidungsfindung in der kommunalen Planung hinsichtlich der Bedürfnisse der Babyboomer datenbasiert unterstützen soll.

Die Anforderungserhebung erfolgte in einem dreistufigen Prozess, der Voranalysen, Anforderungsworkshops mit sieben ländlichen, suburbanen bzw. urbanen Modellkommunen sowie Analyseworkshops mit Forschenden des Konsortiums umfasste. Die Anforderungen sind in zwei Nutzungsszenarien zur bedarfsgerechten Analyse und Planung 1) nachhaltiger Wohnformen sowie 2) kommunaler Angebote beschrieben.

Die Umsetzung des DSS erfolgt in Form einer Client-Server-Architektur. Kernfunktionalitäten beinhalten neben der Visualisierung kommunaler Basisdaten (u. a. Soziodemografie, Finanzdaten, Befragungsdaten) die geodatenbasierte Darstellung verschiedener Versorgungsindikatoren (u. a. Gesundheitsangebote, soziale Einrichtungen, Nahversorgung), die Analyse der Versorgungsqualität in Quartieren mittels eines Tools zur intermodalen Routenplanung sowie die Szenario-basierte Optimierung von Versorgungsstandorten und alternativen Angebotsformen.

Aktuell befindet sich der Test des Ageing Smart DSS in den sieben Modellkommunen in Vorbereitung. Im Fokus des weiteren Projektverlaufs stehen die Umsetzung des Nutzer:innenfeedbacks sowie die Frage, in wie weit das DSS auf weitere Nutzungsszenarien (z. B. Bedürfnisse weitere Zielgruppen kommunaler Planung wie Kinder und Jugendliche) übertragen werden kann.

[1] Fraunhofer IESE, Smart City Design, Fraunhofer-Platz 1, 67663 Kaiserslautern,
matthias.berg@iese.fraunhofer.de, https://orcid.org/0000-0001-8532-8651
[2] Fraunhofer IESE, Smart City Engineering, Fraunhofer-Platz 1, 67663 Kaiserslautern,
benjamin.hassenfratz@iese.fraunhofer.de, https://orcid.org/0000-0003-4302-0821

doi:10.18420/BTW2025-109

Literaturverzeichnis

[AH24] Abu Qasem, R.; Hausbrand, N.: Decision Support System for Inclusive Smart City Design: Baby Boomers as a use Case. In ERCIM News 135, 2024, S. 31-32.

[SB23] Statistisches Bundesamt: Endgültige Kinderzahl (je Frau): Deutschland, Geburtsjahr der Frau, Altersgruppen der Frauen. Online: https://www-genesis.destatis.de/datenbank/online/url/6a5ac33f. Zugegriffen am 13.12.2024.

App LitSpatz for literary walks to promote primary students' text comprehension, perspective taking and related skills: Experiences, findings and implications

Nora Heyne [1], Maximilian Pfost [2], and Peter Kuntner [3]

Abstract: The teaching of listening and reading skills, including abilities to adopt spatial and affective perspectives of literary figures and to reflect critically on texts with regard to one's own lifeworld, are central goals of primary school education in Germany [KM22]. However, many primary school students fail to achieve minimum standards in these skills [ML23; SS22]. To support students in developing these skills and to record their current skill levels, the LitSpatz was developed. The app is based on previous findings on children's academic, cognitive and affective-motivational development as well as on theoretical approaches and results on text-picture-comprehension [LB07; Ma21] and reading instruction [WN12; WJ12]. Within the pilot version of the app, primary school students are offered the opportunity to take part in literary walks through the city of Bamberg (Germany), either on- or off-site. Therein, a walk-in story is presented auditorily and illustrations or references to authentic locations are displayed to visualize the scenes´ settings. Furthermore, cognitively activating and further questions are asked. The presentation provides an introduction of the implementation of the app in both versions for the participation: a) in the city of Bamberg (on site) and b) for virtual literary walks (off-site) in reading classes. Moreover, it shows participation rates and selected results of previous participation. In particular, outcomes on students´ characteristics and abilities in relation to text comprehension and aspects of perspective taking (spatial and affective perspective taking, transportation, empathic text comprehension) are presented. Concerning low participation rates found for the on site version, assumed causes and impact factors are discussed. Finally, results from the previous use of the app are discussed with respect to future steps regarding the improvement of the offer as well as its enrichment by implementing further geospatial technologies.

Keywords: walk-in story, primary students, listening comprehension, reading comprehension, spatial perspective taking

[1] University of Bamberg, Department of Educational Research, Markusplatz 3, Bamberg, 96045, nora.heyne@uni-bamberg.de, https://orcid.org/0000-0002-5029-3994
[2] University of Bamberg, Department of Educational Research, Markusplatz 3, Bamberg, 96045, maximilian.pfost@uni-bamberg.de, https://orcid.org/0000-0002-7066-0456
[3] University of Bamberg, Department of Educational Research, Markusplatz 3, Bamberg, 96045, peter.kuntner@uni-bamberg.de https://orcid.org/0000-0001-9950-6340
[3] University of Bamberg, Department of Educational Research, Markusplatz 3, Bamberg, 96045.

doi:10.18420/BTW2025-110

Bibliography

[KM22] KMK, Kultusministerkonferenz, https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2022/2022_06_23-Bista-Primarbereich-Deutsch.pdf, *Bildungsstandards für das Fach Deutsch Primarbereich* [Educational standards for the subject German at primary level: Resolution of the Standing Conference of the Ministers of Education and Cultural Affairs]

[LB07] Liben, L. S., & Myers, L. J. (2007). Developmental changes in children's understanding of maps: What, when, and how? In J. M. Plumert & J. P. Spencer (Eds.), *The emerging spatial mind* (pp. 193–218). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195189223.003.0009

[Ma21] Mayer, R. E. (2021). Cognitive Theory of Multimedia Learning. In R. E. Mayer & L. Fiorella (Eds.), *The Cambridge Handbook of Multimedia Learning* (3rd ed., pp. 57–72). Cambridge University Press. https://doi.org/10.1017/9781108894333.008

[ML23] McElvany, N., Lorenz, R., Frey, A., Goldhammer, F., Schilcher, A., & Stubbe, T. C. (Eds.). (2023). *IGLU 2021: Lesekompetenz von Grundschulkindern im internationalen Vergleich und im Trend über 20 Jahre*. Waxmann. https://www.waxmann.com/index.php?eID=download&buchnr=4700

[SS22] Stanat, P., Schipolowski, S., Schneider, R., Sachse, K. A., Weirich, S., & Henschel, S. (Eds.). (2022). *IQB-Bildungstrend 2021: Kompetenzen in den Fächern Deutsch und Mathematik am Ende der 4. Jahrgangsstufe im dritten Ländervergleich* [IQB Education Trend 2021: Competencies in the subjects German and mathematics at the end of year 4 in the third country comparison]. Waxmann.

[WN12] Wilkinson, I. A. G. & Nelson, K. (2012). 7.7 Role of Discussion in Reading Comprehension, In: J. Hattie & E.M. Anderman (Eds.), *International guide to student achievement* (pp. 299–302). Taylor & Francis Group.

[WJ12] Wilson, A., Jesson, R., & Cnaughton, S. M. (2012). 7.18 Reading – Vocabulary Programs, In: J. Hattie & E.M. Anderman (Eds.*), International guide to student achievement* (pp. 332–334). Taylor & Francis Group.

Smart Cities as enablers of collaborative open-source software development

Christian Weidner [1]

Abstract: This poster explores the impact of smart cities on digital sovereignty and the public sector's engagement in governmental open-source software (OSS) projects. A quantitative survey of OSS community members found significant correlations between higher Smart City Index (SCI) scores and greater participation in supporting tasks such as bug reporting and feature requests. Such engagement fosters innovation and strengthens collaboration within digital public infrastructure. These findings underline the need for further research on how smart cities promote openness, participation, and digital literacy among citizens and public servants.

Keywords: Open-source software, Smart City, Local Government, Public Sector, Collaboration

Smart cities (SC) are characterized by openness and collaboration [Pa11, ABD15, Gr23]. In this context, OSS plays a crucial role in various SC software projects [Ju19]. A SC facilitates efficient knowledge flows, reduced transaction costs, making it "optimized for [open] innovation activities" [Ha24]. OSS governance highlights the significance of large and diverse communities, as described by Linus' Law [Ra99, MW09].

As part of a case study [Yi18] on German geoportal OSS *Masterportal*, an online survey was conducted among members of a contractual network of 49 public agencies securing Masterportal's funding and development [We25], representing a heterogeneous smart city landscape [DSN21]. Using a 7-point Likert scale a total of 38 participants rated transaction characteristics [Wi81], as well as their supporting and active participation [CG19]. The SCI 2024, published by *Bitkom*, served as a benchmark for smart city maturity [Bi24, La24]. Four *Länder* and nine *Kreise* were matched to an SCI score, totaling 13 cases (Fig. 1). Multiple linear regression was conducted using *R* [R 22] and *Jamovi* [Th24].

Higher SCI scores were associated with greater participation in accessible supporting tasks, particularly within local governments (Fig. 2). However, correlations between increased SCI scores and code contributions were notably weaker. Given the relatively small sample size, the single-case study design, reliance on a single indicator, and the focus solely on perceived participation, these findings are clearly limited. Participants may have over- or underestimate their organization's involvement or interpret the Likert scale inconsistently. Nevertheless, the identified correlations warrant further investigation, for instance, through monitoring activities in software repositories and more precise benchmarks for SC maturity than SCI is able to provide. The smart city discourse should place greater emphasis on institutional factors that shape openness as a key concept, particularly in terms of digital skills, organizational culture, and normative expectations.

[1] Fraunhofer Institute for Open Communication Systems FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany, christian.weidner@fokus.fraunhofer.de, https://orcid.org/0009-0009-7798-008X

doi:10.18420/BTW2025-111

Bibliography

[ABD15] Albino, Vito; Berardi, Umberto; Dangelico, Rosa Maria: Smart Cities: Definitions, Dimensions, Performance, and Initiatives. Journal of Urban Technology, 22(1):3–21, 2015.

[Bi24] Bitkom: Smart City Index 2024. Ausführliche Ergebnisse. Technical report, Bitkom e. V., Berlin, 2024.

[CG19] Cheng, Jinghui; Guo, Jin L.C.: Activity-Based Analysis of Open Source Software Contributors: Roles and Dynamics. In: 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). Montréal, QC, Canada, pp. 11–18, 2019.

[DSN21] Degkwitz, Till; Schulz, Daniel; Noennig, Jörg Rainer: Cockpit Social Infrastructure: A Case for Planning Support Infrastructure. International Journal of E-Planning Research (IJEPR), 10(4):104–120, 2021.

[Gr23] Gracias, Jose Sanchez; Parnell, Gregory S.; Specking, Eric; Pohl, Edward A.; Buchanan, Randy: Smart Cities—A Structured Literature Review. Smart Cities, 6(4):1719–1743, 2023.

[Ha24] Han, Junghee: Open innovation in a smart city context: the case of Sejong smart city initiative. European Journal of Innovation Management, ahead-of-print(ahead-of-print), 2024.

[Ju19] Jussila, Jari; Kukkamäki, Joni; Mäntyneva, Mikko; Heinisuo, Juuso: Open Data and Open Source Enabling Smart City Development: A Case Study in Häme Region. Technology Innovation Management Review, 9(9):25–34, 2019.

[La24] Lange, Felix: Hinweise zur Methodik. Technical report, Bitkom e. V., Berlin, 2024.

[MW09] Meneely, Andrew; Williams, Laurie: Secure open source collaboration: an empirical study of linus' law. In: Proceedings of the 16th ACM conference on Computer and communications security. CCS '09, New York, NY, USA, pp. 453–462, 2009.

[Pa11] Paskaleva, Krassimira Antonova: The smart city: A nexus for open innovation? Intelligent Buildings International, 3(3):153–171, 2011.

[R 22] R Core Team: R: A Language and Environment for Statistical Computing, 2022. Version 4.2.2.

[Ra99] Raymond, Eric: The cathedral and the bazaar. Knowledge, Technology & Policy, 12(3):23–49, 1999.

[Th24] The jamovi project: jamovi, 2024. Version 2.5.6.

[We25] Weidner, Christian: Governance von Open-Source-Software im öffentlichen Sektor: Make, Buy or Contribute? In: Tagungsband FOSSGIS-Konferenz 2025. Münster, Germany, 2025.

[Wi81] Williamson, Oliver E.: The Economics of Organization: The Transaction Cost Approach. American Journal of Sociology, 87(3):548–577, 1981.

[Yi18] Yin, Robert K.: Case study research and applications: design and methods. SAGE, Los Angeles, CA, 6th edition, 2018.

Appendix

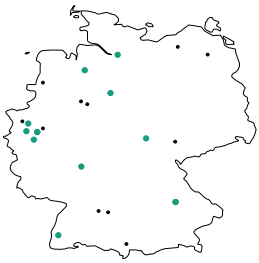


Fig. 1: Distribution of survey participants, large green dots with allocable SCI value, small black dots represent remaining participants.

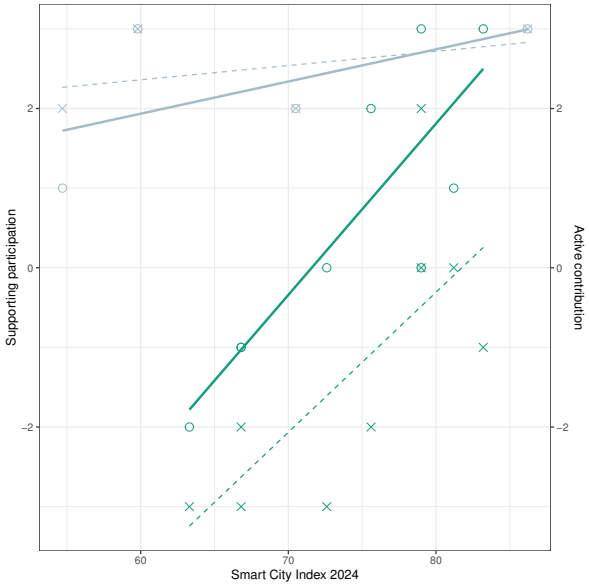


Fig. 2: Scatter plot of multiple linear regression. Green represents districts (*Kreise*), gray represents states (*Länder*). Circles and solid reg. lines: ρ(SCI score, supporting participation), cross markers and dashed reg. lines: ρ(SCI score, active contribution).

Gender-sensitive urban planning: Connections between gender, perception of safety in public space, and environmental design

Katharina Helgerth [1] and Astrid Schütz [1]

Keywords: Gender, Urban Planning, Safety Perception

Abstract

There is a growing trend among cities around the world to integrate sustainability considerations into their strategic agendas. However, the UN Women's report from 2017-2020 points at persistent gaps in achieving the United Nations' 2030 Agenda and its 17 Sustainable Development Goals (SDGs), particularly SDG 5, which aims for gender equality, and SDG 11, which focuses on creating safe, inclusive, and sustainable urban environments. Despite these objectives, the safety and inclusion requirements of women and gender-diverse individuals remain largely unmet. Contemporary urban safety strategies frequently focus on objective risks over subjective safety, thereby failing to acknowledge the substantial influence of perceived safety on individuals' behaviours and their quality of life. This study addresses this research gap by examining the relationship between environmental factors and subjective safety in a medium-sized town in Germany, Bamberg. It investigates the impact of environmental factors on perceptions of safety in the city's public spaces. The methodology will comprise a spatial analysis using virtual mapping of locations that are perceived as unsafe. The findings are intended to inform urban planning practice by identifying gender-specific safety needs and promoting more inclusive and sustainable urban development.

[1] University of Bamberg, Chair of Personality Psychology and Psychological Assessment, Germany, katharina.helgerth@uni-bamberg.de, https://orcid.org/0009-0003-9679-8263; astrid.schuetz@uni-bamberg.de, https://orcid.org/0000-0002-6358-167X

doi:10.18420/BTW2025-112

Workshop on Advances in Cloud Data Management

BTW 2025 Workshop on Advances in Cloud Data Management

Jana Giceva[1], Martin Hentschel[2] und Tobias Ziegler[3]

Abstract: The BTW 2025 Workshop on Advances in Cloud Data Management explored recent developments and future directions in cloud data management. Speakers discussed advancements in data warehousing, query optimization, and data pipeline architectures to enhance performance and efficiency in cloud environments. The workshop also addressed challenges in managing data for modern software architectures, such as event-driven microservices, and the complexities of decomposing database systems. Additionally, presentations covered the potential of serverless computing for cost-efficient data processing and the importance of fine-grained access control for data governance.

We want to thank all speakers for delivering highly interesting talks and for making this workshop a success.

1 MotherDuck: DuckDB backed by the cloud

Peter Boncz, MotherDuck and CWI Amsterdam

MotherDuck is a new service that connects DuckDB to the cloud. It introduces the concept of "dual query processing: the ability to execute queries partly on the client and partly in the cloud. The talk covers the motivation for MotherDuck and some of its use cases; as well as the main characteristics of its system architecture, which heavily uses the extension mechanisms of DuckDB. To provide context, the talk will therefore also provide a brief overview of the DuckDB architecture. The talk will also cover ongoing research work related to MotherDuck in the area of caching as well as query optimization.

2 The Future of the Cloud and the Future of Cloud Databases

Viktor Leis, TU München

Cloud computing is transforming the technology landscape, with database systems at the forefront of this change. A striking example is an online bookstore that has grown to dominate the database market. The appeal of cloud computing for IT users lies in several key factors: a reduced total cost of ownership through economies of scale and advanced services that minimize the burden of ündifferentiated heavy lifting". More broadly, cloud computing reflects a civilizational trend toward increased technological and economic specialization.

[1] TU München, jana.giceva@in.tum.de
[2] IT University of Copenhagen, mhent@itu.dk
[3] TU München, t.ziegler@tum.de

doi:10.18420/BTW2025-113

However, the current state of cloud computing often falls short of these promises. Hyperscalers are evolving into vertically integrated oligopolies, controlling everything from basic server rentals to high-level services. This trend is only accelerating, potentially leading to a future where hyperscalers establish software standards and design their own hardware, making it impossible to compete. Moreover, despite differences in branding, the major cloud providers are fundamentally similar, lacking interoperability and fostering vendor lock-in. As a result, we risk returning to the monopolistic conditions of the IBM and Wintel eras and ultimately technological stagnation due to limited competition.

Yet there is cause for optimism. Great technology can still succeed, as the multi-cloud data warehouse Snowflake has shown. The rise of data lakes and open standards, such as Parquet and Iceberg, further underscores the potential for interoperability and innovation. Additionally, there are orders-of-magnitude gaps between the price of existing cloud services and what is theoretically achievable, creating opportunities for disruption. These price gaps persist because cloud services are inherently complex to build, requiring redundant efforts and leading to high barriers to entry. For example, a DBMS might need a highly available control plane, a write-ahead log service, and distributed storage servers. None of these abstractions is available as a read-to-use service, which makes it difficult to enter the cloud database market. The current cloud landscape is more a result of historical circumstances than optimal design, leaving ample room for disruption.

In this talk, I will outline a blueprint for reinventing the cloud by focusing on three key areas: First, we need a unified multi-cloud abstraction over virtualized hardware. Second, we should establish new open standards for existing low-level cloud services. Third, we need abstractions that simplify the creation of new cloud services, such as reusable control planes and foundational components like log services and page servers. Together, this will make it significantly easier to build, deploy, and monetize new cloud services. Increased competition would commoditize foundational services and spur technological innovation.

3 Firebolt Transactions: Consistency, Performance and Availability - Pick All Three

Benjamin Wagner, Firebolt

Firebolt is a data warehouse built for data intensive applications. To support these workloads, our metadata services enable:

- An unlimited number of concurrent writers across a region
- Strong consistency with snapshot isolation
- Low overhead for read-only transactions (2ms) on petabytes of data
- Powerful metadata operations such as zero-copy cloning and time travel

This talk provides a deep-dive into how we built Firebolt's metadata services on top of FoundationDB. We focus on how to leverage the underlying key-value space in a way that supports low-latency transactions. Based on this, we describe our internal API design as well as dependent services such as metadata snapshot compaction and garbage collection. Finally, we describe how we deploy our service on AWS to minimize network latency.

4 Preparing Data for Analytics: Exploring Modern Approaches to Data Pipelines

Fabian Hueske, Confluent

Cloud data warehouses and data lakes power the analytical workloads of many enterprises. These systems store vast amounts of data, generated by external sources, that must be ingested before they are ready for querying. The ingested data typically requires cleaning, transformation, enrichment, integration, and aggregation to ensure it is in the right format for effective analysis.

Given the scale of data being processed, the transformation engines responsible for these tasks must offer high throughput while maintaining cost efficiency. Furthermore, low-latency processing is important for meeting the demands of many real-time use cases.

Different architectures exist for implementing data pipelines that perform these transformations. Some, like Snowflake's Dynamic Tables, rely on periodic batch processing, while others leverage stateful stream processing engine such as Apache Flink. In this talk, we discuss different data pipeline architectures and analyze their strengths, limitations, and trade-offs.

5 Workload-Driven Indexing in the Cloud

Andreas Kipf, TU Nürnberg

In this talk, I will present predicate caching, a lightweight secondary indexing mechanism for cloud data warehouses. Specifically, I will show that workloads are highly repetitive, i.e., users and systems frequently send the same queries. To improve query performance on such workloads, most systems rely on techniques like result caching or materialized views. However, these caches are often stale due to inserts, deletes, or updates that occur between query repetitions. Predicate caching, on the other hand, improves query latency for repeating scans and joins in a lightweight manner, by simply storing ranges of qualifying tuples. Such an index can be built on the fly and can be kept online without recomputation. We implemented a prototype of this idea in the cloud data warehouse Amazon Redshift. Our evaluation shows that predicate caching improves query runtimes by up to 10x on selected queries with negligible build overhead.

6 OBSERVE - Petascale Streaming for Observability

Tomas Karnagel, Observe

Observe brings together petabyte-per-day streaming ingest, relational analytics, search, and real-time monitoring capabilities under one product. Observe was built to enable all types of observability workloads — logs, metrics, traces, application performance, and security — as well as complex business data analytics, over a single connected data lake. The platform is powered by the Snowflake Data Cloud supporting our hundreds of millions of queries per day. In this talk we will give an overview of the architecture and capabilities of the Observe platform.

7 Query acceleration via auto-tuning in Amazon Redshift

Panos Parchas, AWS Redshift

Amazon Redshift is the first fully-managed, petabyte-scale, enterprise-grade cloud data warehouse that revolutionized the data warehousing industry. During the last decade, the Redshift team is constantly innovating by extending the functionality and improving the efficiency of the system. A large focus area has been ëase of use"that targets on auto-tuning and ML techniques to make the system more performant for the unique characteristics of individual workloads. This talk provides an overview of Redshift's architecture, focusing on query processing. Within this context, we discuss techniques that the team has developed during the past couple of years for query acceleration and we dive deep into our novel data distribution and data layout schemes.

8 The Fine Art of Work Skipping

Ismail Oukid, Snowflake

Modern cloud-based analytics systems may have to process petabytes of data per query. The most efficient way to process this data is to not process it at all, i.e., to skip work. The most common work skipping technique is Pruning, a family of techniques that helps skip loading and processing data that does not pertain to the final result. In this talk we will discuss why pruning is so important for query performance, especially in a cloud-based analytical system, by analyzing Snowflake customer workloads. We will explore various pruning techniques employed at Snowflake - filter pruning, TopK pruning, and join pruning - and demonstrate how their combined application skips the majority of micro-partitions. We will conclude by briefly touching on another type of work skipping, namely result caching and reuse.

9 Data Management in Event-Driven Microservice Architectures

Yongluan Zhou, Copenhagen University

Building cloud-native applications necessitates new approaches to software architecture to achieve a high level of scalability, elasticity, responsiveness, fault tolerance, and decoupling. Event-driven microservice architecture (EDMA) emerges as a suitable architectural style that fulfills this requirement. EDMA encourages the breakdown of an application into independent and asynchronous components that can be deployed, scaled, and evolved separately while allowing for the isolation of failures from one another. The growing popularity of EDMA in the industry has prompted cloud providers to offer rich features tailored for its deployments, such as specialized container-based technologies for deploying and scaling microservices, message queueing services for communication between loosely coupled microservices, multi-tenant database technologies to support the isolation of microservices, and various application frameworks and side-car technologies to facilitate code development, evolution, and maintenance of microservices.

However, due to the asynchronous nature of EDMA, event-driven microservices often adopt eventual consistency following the BASE model. Our recent survey found that these practices lead to many data management challenges in achieving various application safety properties. Essentially, EDMAs sacrifice the important benefits of traditional n-tier architectures: completely delegating data management, failure recovery, and data consistency assurances to the database systems. On the contrary, developers are burdened with implementing these features within the application code. These challenges have sparked the recent calls to move away from EDMA and revert to the traditional n-tier architecture. In this talk, we will argue that it is feasible to evolve data management systems to deliver advantages from both worlds. A fundamental issue is that the decades-old database programming abstraction, which includes database programming APIs (such as JDBC) and stored procedures, does not meet the demands of modern software architectures like EDMA. Modernizing the programming abstraction and system architecture of database systems is the key to achieving this goal.

10 The challenges of decomposing database systems in the cloud

Alexander Böhm, SAP

Modern cloud native software architectures follow a micro-services approach. They decompose complex applications into sets of small, individual services with clearly defined APIs and can be implemented by small development teams. Ideally, these microservices can iterate quickly, with short development cycles, frequent releases to production, a small blast radius in case of failures, and high degrees of freedom regarding e.g. the choice of the programming language and development style. Moreover, the individual services can be scaled separately, leading to a better, more fine-grained resource allocation and reduced overall costs.

Cloud-native database management systems such as Aurora, AlloyDB, Sokrates, PolarDB, Spanner, BigQuery, HANA Cloud, and others have recognized this trend, and decomposed their database core into multiple building blocks. Most prominent is the separation into distinct compute and storage layers, but more advanced and nuanced deployments are also found: This includes the XLOG service that factorizes out WAL processing in Sokrates, the disaggregated shuffle layer for in-memory joins in Dremel's runtime system, Spanner's zonemaster data placement service, or the separation of the query optimizer to a separate service in Greenplum's Orca design.

While the overall benefits of decomposition such as better scalability, elasticity, and the efficient use of resources are typically advertised publicly in corporate blogs and academic publications, decomposition also entails notable downsides that are not prominently discussed and often overlooked.

In this talk, we highlight the challenges of decomposing cloud-native database management systems into multiple services using existing industry systems as concrete examples. We also give a perspective on how those challenges can be addressed in a systematic manner. Among others, we discuss the implications of decomposition on latency, which is particularly important for transaction processing and HTAP systems such as Aurora, AlloyDB, Sokrates, and HANA Cloud.

We outline the additional complexity for troubleshooting highly distributed systems with potentially dozens of services, and how this challenge can be addressed. Moreover, we review the implications of separating tightly coupled components (e.g. the query optimizer, metadata catalog, and runtime system). We conclude our overview with a discussion of the consequences of using (too) many microservices for the availability and reliability of the overall database management system, and highlight implications for the development processes of the involved services and teams.

11 Data Processing on Elastic Cloud Resources

Thomas Bodner, HPI Potsdam

Analytical data products, such as business intelligence reports and machine learning models, require processing large amounts of data using extensive computational resources. Traditionally, provisioning resources involves high up-front expenses. The cloud, as a short-term provisioning model, provides cost-effective access to pools of resources and, as a result, is the standard for deploying data processing systems today. Recently, serverless cloud computing embodies resource pools that are highly elastic. This elasticity has the potential to make cloud-based systems easier to use and more cost-efficient, avoiding complex resource management and under-utilization.

Motivated by the potential impact that serverless cloud infrastructure has on data processing systems, in this talk we explore the use of this category of highly elastic cloud resources. We

first evaluate the performance and cost characteristics of the public serverless infrastructure from AWS. Based on comprehensive experiments with a range of compute and storage services, as well as end-to-end analytical workloads, we identify distinct boundaries for performance variability in serverless networks and storage. In addition, we find economic break-even points for serverless versus server-based storage and compute resources. These insights guide the usage of serverless infrastructure for data processing.

We then present Skyrise, a query processor that is built entirely on serverless resources. Skyrise employs a number of adaptive and cost-based techniques to operate within the limits, where serverless data processing remains practical. Our evaluation shows that Skyrise provides competitive performance and cost with commercial Query-as-a-Service (QaaS) systems for terabyte-scale queries of analytical TPC benchmarks. Furthermore, Skyrise leverages the elasticity of its underlying infrastructure for cost efficiency in ad-hoc and low-volume workloads, compared to cloud data systems deployed on virtual servers.

Overall, we show that serverless resources are a viable foundation and offer economic gains for data processing. Since current serverless platforms have various limitations, we discuss how our results can be extended to emerging serverless system designs.

12 Databricks Lakeguard: Supporting fine-grained access control and multi-user capabilities for Apache Spark workloads

Sebastian Hillig, Databricks

Enterprises want to apply fine-grained access control policies to manage increasingly complex data governance requirements. Rich policies should be uniformly applied across all their workloads. In this paper, we present Databricks Lakeguard, our implementation of a unified governance system that enforces fine-grained data access policies, row-level filters and column masks across all of an enterprise's data and AI workloads. Lakeguard builds upon two main components: First, it uses Spark Connect, a JDBC-like execution protocol, to separate the client application from the server and ensure version compatibility. Second, it leverages container isolation in Databricks' cluster manager to securely isolate user-supplied code from the core Spark engine. With Lakeguard, a user's permissions are enforced for any workload and in any supported language, SQL, Python, Scala, and R on multi-user compute. This work overcomes fragmented governance solutions, where fine-grained access control could only be enforced for SQL workloads, while big data processing with frameworks such as Apache Spark relied on coarse-grained governance at the file level with cluster-bound data access.

13 Decluttering the data mess in LLM training

Maximilian Böther, ETH Zürich

Training large language models (LLMs) presents new challenges for managing training data due to ever-growing model and dataset sizes. State-of-the-art LLMs are trained over trillions of tokens that are aggregated from a cornucopia of different datasets, forming collections such as RedPajama, Dolma, or FineWeb. However, as the data collections grow and cover more and more data sources, managing them becomes time-consuming, tedious, and prone to errors. The proportion of data with different characteristics (e.g., language, topic, source) has a huge impact on model performance.

In this abstract paper, we present three challenges we observe for training LLMs due to the lack of system support for managing and mixing data collections. Based on those challenges, we are building Mixtera, a system to support LLM training data management

14 High-Performance Query Processing on Cloud Object Stores

Dominik Durner, CedarDB

The growing adoption of cloud-based data systems is making data management more flexible, scalable, and cost-effective. With virtually unlimited capacity and strong durability guarantees, cloud object storage is becoming essential to modern analytical database systems. This talk focuses on the efficient use of disaggregated cloud object storage for both analytical processing and hybrid transactional and analytical processing (HTAP).

Our work on cloud object storage explores the closing performance gap between network and local NVMe bandwidths, making direct processing on disaggregated cloud storage feasible for many analytical workloads. With the insights from our in-depth study on the economics and performance characteristics of cloud object stores, we developed AnyBlob, a multi-cloud download manager that optimizes high-throughput data retrieval while minimizing CPU overhead. By seamlessly integrating cloud object storage with database query engines, AnyBlob achieves retrieval performance comparable to systems that process data from fast local NVMe SSDs. Overall, we show that processing data from cloud object storage is a viable choice for analytical workloads without sacrificing elasticity or resource efficiency.

Extending this work, we present Colibri, a hybrid column-row storage engine designed to address the specific requirements of HTAP workloads. Colibri separates hot and cold data to support transactional and analytical workloads within a single system. Frequently updated transactional data is stored in an uncompressed row-based format, while analytical data resides in a compressed columnar layout optimized for efficient analytics. To take full advantage of the underlying hardware, Colibri minimizes logging overhead and integrates seamlessly with modern buffer managers. By combining the benefits of AnyBlob's high-throughput cloud integration with Colibri's hybrid storage architecture, we achieve considerable performance improvements on hybrid workloads. Moreover, our high-performance hybrid storage engine enables the elimination of traditional ETL pipelines that introduce high latency and data duplication.

In summary, our work bridges the gap between transactional and analytical processing in the cloud by providing a unified architecture that provides superior performance while being cost-effective.

Workshop on Big (and Small) Data in Science and Humanities

Fifth Workshop on Big (and Small) Data in Science and Humanities (BigDS 2025)

Andreas Henrich[1], Naouel Karam [2], Birgitta König-Ries[3], Richard Lenz[4], Stefanie Scherzinger[5], Bernhard Seeger[6]

The importance of data has dramatically increased in almost all scientific disciplines over the last decade, e.g., in meteorology, genomics, complex physics simulations, biological and environmental research, medicine, and recently also in the humanities and social sciences. This development is due to great advances in data acquisition and accessibility, e.g., improvements in remote sensing, powerful mobile devices, popularity of social networks, and the ability to handle unstructured data (including texts). On the one hand, the availability of such data masses leads to a rethinking in scientific disciplines on how to extract useful information and foster research. On the other hand, researchers feel lost in the data masses because appropriate data management, integration, discovery, analysis, and visualization tools are only rudimentarily available so far. However, this is starting to change with the recent developments of big data technologies and with progress in AI, natural language processing, semantic technologies, and others that are not only useful in business, but also offer great opportunities in science and humanities. Scientific workflows must be realized as flexible end-to-end analytic solutions to allow for complex data processing, integration, analysis, and visualization of Big Data in various application domains.

For the workshop, we were particularly interested in two aspects: First, how can tools support the achievement of the FAIR principles, e.g., legal aspects? And second, what contributions can the database and information systems community make to the conceptualisation and implementation of Germany's National Research Data Infrastructure NFDI[7]?

This workshop intended to bring together scientists from various disciplines and NFDI consortia with database researchers to discuss real-world problems in data research in-

[1] University of Bamberg, Media Informatics, 96047 Bamberg, Germany andreas.henrich@uni-bamberg.de
[2] Institut für Angewandte Informatik (InfAI), 04109 Leipzig, Germany karam@infai.org
[3] University of Jena, Heinz Nixdorf Chair for Distributed Information Systems, 07743 Jena, Germany birgitta.koenig-ries@uni-jena.de
[4] Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Informatik 6, 91058 Erlangen richard.lenz@fau.de
[5] Universität Passau, Chair for Scalable Database Systems, 94032 Passau stefanie.scherzinger@uni-passau.de
[6] University of Marburg, Department of Mathematics and Computer Science, 35032 Marburg, Germany seeger@informatik.uni-marburg.de
[7] https://www.dfg.de/en/research_funding/programmes/nfdi/index.html

doi:10.18420/BTW2025-114

frastructures, data science, and big data technologies. We are happy that the submissions reflected this goal with several articles with strong links to NFDI consortia being among them.

This is the fifth edition of the workshop on Big (and Small) Data in Science and Humanities (BigDS) after editions at BTW 2015, 2017, 2019, and 2023. This edition of the BigDS workshop again is co-located with the Conference on Database Systems for Business, Technology and Web (BTW). With workshop contributions from various disciplines, we hope to promote the dialogue between domain experts and data scientists and to foster the engagement of the database community to NFDI and other important infrastructure projects.

The workshop program kicks off with a joint keynote with the DE4DS workshop. Laura Koesten from the University of Vienna and the Mohamed bin Zayed University of Artificial Intelligence gives an overview over "Perspectives on Human-Data Interaction" in her invited keynote. Laura Koesten's keynote explores different perspectives on Human-Data Interaction, examining how people engage with, experience, and make sense of data. Her research aims to increase transparency in the decisions that shape data—spanning collection, analysis, communication, and reuse—while addressing the challenge of explaining both data and models to diverse audiences. In her talk, she presents insights from two key research areas: data-centric sensemaking, which frames data as a dynamic process, and the interpretation of visualizations, with a focus on how messages shape audience understanding. Drawing on her ongoing project *Talking Charts*, she illustrates these concepts with examples from research involving both expert and general audiences.

We further selected eight contributions that address different challenges in the context of data-driven processing and analytics.

The first session ends with a paper on XCIP, a component designed to address non-functional requirements in data integration services for research data infrastructures like NFDI. XCIP balances provider needs such as access control, resource usage limitations, and usage statistics with user expectations for scalable and fast data access.

The second session is focused on topics related to semantic web technologies and their role in providing FAIR data, a subject at the heart of many NFDI consortia. Two papers explore ontology-driven data acquisition. While Caspar Felix Hanika and co-authors propose an approach allowing non-experts to adapt and extend ontologies that do not quite meet their needs, Philipp Stangl et al. show a use-case from chemistry where ontology-driven data acquisition supports the provision of data for machine-learning based catalyst and reaction design. The latter has been investigated in the context of NFDI4CAT. Philipp Plamper and colleagues have compiled an overview of ontologies relevant for research in fields related to global change in particular ecosystem, biodiversity, and agricultural research. The work by Al Mustafa et al. addresses a similar domain and describes the development of a knowledge graph and a query builder to support non-expert access to it integrating different types of plant-related data.

Fillies and coauthors describe efforts that are part of NFDI4Biodiversity. They aim to solve challenges in dealing with historic species names. They propose an LLM based solution to map these terms (both common and scientific) to current nomenclature. Usage of LLMs is also investigated by Jens Bruchertseifer et al. in their contribution. They explore whether a zero-shot classification pipeline is useful to assign topic-labels to scientific papers and evaluate this with DBLP data.

Leon Fruth and his coauthors aim to support work in the Digital Humanities with a generic search system that allows transparent access across various heterogeneous authority files. This work is part of the Text+ NFDI endeavor. We want to thank everyone who contributed to the workshop, especially the authors, the keynote speaker Laura Koesten, the BigDS program committee, the BTW team, and all the participants.

Workshop Organizers

Andreas Henrich (Univ. Bamberg)
Naouel Karam (InfAI)
Birgitta König-Ries (Univ. Jena)
Richard Lenz (Univ. Erlangen-Nürnberg)
Stefanie Scherzinger (Univ. Passau)
Bernhard Seeger (Univ. Marburg)

Program Committee

Alsayed Algergawy (Univ. Passau)
Thomas Brinkhoff (Jade Hochschule)
Stefan Deßloch (RPTU Kaiserslautern)
Jana Diesner (TU München)
Thomas Eckart (Univ. Leipzig)
Michael Gertz (Univ. Heidelberg)
Nikolaus Glombiewski (Univ. Marburg)
Anton Güntsch (Botanischer Garten und Botanisches Museum, Berlin)
Andreas Hardt (FAU Erlangen-Nürnberg)
Alfons Kemper (TU München)
Toralf Kirsten (Univ. Leipzig)
Ulf Leser (HU Berlin)
Bertram Ludäscher (University of Illinois at Urbana-Champaign)
Wolfgang Müller (HITS, Heidelberg)
Christoph Neumann (OTH Amberg)
Thorsten Papenbrock (Univ. Marburg)
Matthias Renz (CAU Kiel)

Harald Sack (KIT)
Sirko Schindler (DLR Institut für Datenwissenschaften)
Sonja Schimmler (Fraunhofer FOKUS, Berlin)
Dagmar Triebel (SNSB, München)
York Sure-Vetter (KIT)
Philipp Wieder (Univ. Göttingen)
Claus Weiland (Senckenberg Gesellschaft für Naturforschung, Frankfurt)

The Case for XCIP: Bridging User and Data Provider Expectations in Research Infrastructures

Dominik Brandenstein [1], Christian Beilschmidt [2], Johannes Drönner [2], Nikolaus Glombiewski [1], Michael Mattig [2], and Bernhard Seeger [1]

Abstract: Data integration services are of utmost importance in research data infrastructures like the NFDI, enabling the harmonization of diverse data sources. While significant progress has been made in automatic schema mapping and protocol development, non-functional requirements play a vital role in ensuring fair and transparent collaboration between data infrastructures and providers. Key non-functional requirements include access control, resource usage limitations, quota tracking, and usage statistics reporting. These are crucial for maintaining provider trust and data integrity. For instance, access control ensures that only authorized users can access sensitive data, while usage statistics allow providers to track the impact of their contributions. Balancing these requirements with user experience presents a challenge. End-users expect scalable and fast data access without noticeable restrictions, necessitating a solution that addresses both provider needs and user expectations. To address this challenge, we present the XCIP (eXchange Component for Infrastructure Providers), a novel exchange component designed to facilitate connection and communication between data infrastructures and providers, which builds upon our previous work, the SDExL (Spatiotemporal Data Exchange Layer). XCIP not only addresses non-functional requirements but also introduces innovative mechanisms for data caching and access optimization. These features enhance data accessibility for end-users while maintaining transparency, effectively bridging the gap between provider requirements and user expectations. This paper details the design of XCIP, demonstrating how it addresses non-functional requirements while simultaneously improving the end-user experience through transparent data caching and statistics collection.

1 Introduction

Data integration (DI) is a key feature of research infrastructures (RI) as it allows users to access data from different sources easily without manual integration. Although this is beneficial for users, it poses challenges for data providers. They have to hand over control of their data to DI tools, making it difficult to enforce usage policies and ensure correct attribution. Providers also lose visibility of how (often) their data is being used. These concerns often make providers reluctant to integrate with RIs, despite the potential benefits for both sides and the benefits for users easily accessing data from multiple sources.

[1] University of Marburg, Dept. of Mathematics and Computer Science, Hans-Meerwein-Str. 6, 35032 Marburg, Germany, branden8@informatik.uni-marburg.de, https://orcid.org/0009-0002-1901-9935; glombien@informatik.uni-marburg.de, https://orcid.org/0000-0003-2876-3918; seeger@informatik.uni-marburg.de, https://orcid.org/0000-0002-9362-153X

[2] Geo Engine GmbH, Am Kornacker 68, 35041 Marburg, Germany, christian.beilschmidt@geoengine.de, https://orcid.org/0009-0001-6297-0921; johannes.droenner@geoengine.de, https://orcid.org/0009-0003-9629-2844; michael.mattig@geoengine.de, https://orcid.org/0009-0006-1893-5391

doi:10.18420/BTW2025-115

This paper introduces the concepts behind the eXchange Component for Infrastructure Providers (XCIP), a continuation of our earlier work on the Spatiotemporal Data Exchange Layer (SDExL) [Be23a]. Parts of XCIP are already implemented within the geospatial processing platform Geo Engine [Be23b], which is used as a DI tool within the research data infrastructure Research Data Commons (RDC) [Di23b] of the NFDI4Biodiversity consortium[3]. XCIP is designed to address key issues in data integration within research infrastructures, striking a balance between the needs of data providers and users. The component offers four primary features:

- Data Usage Insights: XCIP tracks and analyzes data utilization patterns, offering data providers valuable information about who uses their data and how frequently it is accessed.
- Enhanced Attribution: The system facilitates accurate data citation, ensuring proper recognition of data providers and owners.
- Performance optimization: By implementing caching strategies, XCIP enhances responsiveness for users while minimizing the load on data providers' systems.
- Access control: Data providers retain control over their data through end-to-end adherence to their access policies within XCIP.

The remaining part of this paper is structured as follows. We first discuss related work. Following that, we present the design of XCIP in detail and then provide practical examples of how we apply it in connection to several data providers within Geo Engine. Finally, we end with our conclusions.

2 Related Work

As discussed in the seminal textbook on data integration [DHI12], state-of-the-art approaches to data integration distinguish between full source materialization within a data warehouse and virtual integration without any data materialization. As outlined in [Ka15], these extremes are not viable for RI. Therefore, XCIP provides a hybrid in which views are partially materialized. The corresponding view selection problem is widely addressed in recent papers [MB12, Ji18]. However, to the best of our knowledge, no prior work exists that takes into account the limitations of data providers on their available sources regarding access control, data delivery limitations, and access statistics.

Data access statistics and user billing are relevant for data marketplaces, e.g., AWS Data Exchange or Snowflake Marketplace. While there is recent work [ZBL23] on pricing considering both the buyer (data users) and the seller side (data providers), these approaches do not consider access rights and limitations of data providers. On the contrary, an RI must consider these limitations when deciding which views to materialize and cache.

[3] https://www.nfdi4biodiversity.org/

Commercial data virtualization platforms like Dremio[4] or Denodo[5] make heavy use of caching and also collect statistics. However, these statistics are not intended for data providers and are not integrated with authorization policies as well as access limitations. They also do not provide approaches to support citations. In general, the problem of an appropriate citation is not well addressed in industry and academia and remains a challenging problem [BDF16].

Our previous work on the so-called SDExL (Spatiotemporal Data Exchange Layer) [Be23a] supports easy connectivity to external data providers via the following three steps. The first step consists of authenticating with the data provider to gain access to its data. The second step is to provide a bijective mapping of external data identifiers to a suitable internal identifier that is unique among different providers and enables a listing of available datasets. The third step is transforming metadata from the external data provider to an internal format and then using this metadata to load the data from the provider. For example, for a satellite image time-series, the provider might specify a spatial extent, resolution, and how multiple files of a dataset relate to each other based on a naming convention with a common prefix followed by the observation time. In case a provider supports an application of filters, SDExL automatically tries to push down a filter to the data provider whenever possible to reduce the amount of data to be transferred from the provider. However, SDExL does not take into account access rights and certain limitations of the provider. It also does not offer an integrated strategy to combine caching and push-down filtering, as recently explored in [Ya21].

3 The Design of XCIP

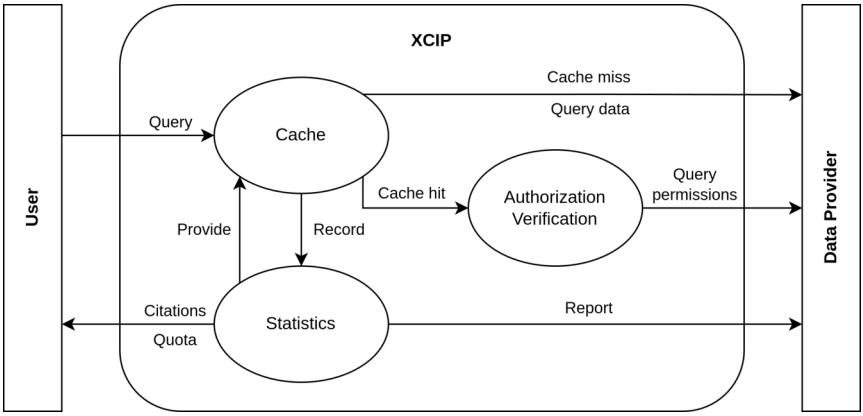


Fig. 1: Overview of XCIP

[4] https://www.dremio.com/
[5] https://www.denodo.com/

The main objective of the design of XCIP is to remain highly flexible and account for the unique requirements and properties of different data providers. At the same time, it offers solutions to overarching problems encountered when connecting to providers. E.g., when it comes to access policies, one provider might heavily restrict access to the offered datasets at an individual user level while another one makes all datasets freely accessible without even requiring user authentication. Or, in regards to the querying capabilities, one provider might only offer geographical feature data through a file server with CSV files while another provider implements an OGC WFS[6] endpoint with a rich query interface. XCIP needs to be able to adapt to these unique features to utilize the providers' existing capabilities best and adhere to their restrictions while abstracting all of these differences away from the user of a system.

XCIP extends and improves on SDExL in multiple aspects. Fig. 1 gives an overview of XCIP's design, where a cache is introduced to improve the performance of user queries and reduce the load on the infrastructures of the providers. As displayed on the left side of Fig. 1, queries from the user are directed to the cache. If the cache does not already contain the queried data, the query is forwarded to the data provider, and the filter pushdown optimization from SDExL is applied, if applicable. Then, the query results can be stored in the cache for future queries. In case the cache already contains the queried data, authorization verification ensures that the corresponding user has sufficient permissions at the involved providers. One of the essential design principles of XCIP is that resource sharing and caching are designed to comply with the access policies of providers without hindering sharing and caching by users with different permissions. In addition, statistics are collected on events such as dataset accesses. These statistics are utilized for quota tracking and usage reports (delivered to the data providers), as well as assisting the user in properly citing the data usage. Moreover, they are used internally in XCIP to make informed cache loading and eviction decisions.

The main features of XCIP are described in detail below. In the following, we introduce our notion of data units, which significantly impacts all other components. The three succeeding subsections provide the details on statistics, authorization, and caching.

As a basis for extending SDExL, we introduce the notion of a *data unit* representing a subset of a dataset from a data provider. Data units describe parts of datasets of arbitrary size. Within a dataset, for example, we could distinguish between individual data records or subsets of records that have a particular property in common, e.g., they are located close together in a geographical region. A data unit identifier then consists of the associated dataset identifier already introduced in SDExL, and, in addition, a filter condition, which uniquely identifies a data unit as a subset of the original dataset. An example would be a concatenation of a DOI (digital object identifier) for the dataset and a filter $year = 2025$ for all records from the year 2025 (e.g., "<DOI>:year=2025"). Thus, given a data unit identifier, XCIP can directly infer the dataset to which it belongs and the corresponding filter. In

[6] https://www.ogc.org/publications/standard/wfs/

addition, for a given filter query posed on a dataset, XCIP can return all the identifiers of the data units in the cache. The data unit identifier differs from other identifiers, e.g., representing metadata, which is not associated with a filter condition but with the whole dataset. In addition, data unit identifiers are used only internally for caching and statistics, while a user can only access the original datasets. As the requirements for caching and statistics can differ, different data units are generally used to meet these requirements better. For example, the data units for caching raster data refer to non-overlapping grid cells with a granularity of 1,000 by 1,000 meters, while a granularity of 100 by 100 meters could be used for statistics.

3.1 Statistics

Statistics are collected based on events that are triggered by an action. An event consists of the triggering action, its timestamp, a list of data unit identifiers on which the action was performed, a unique query identifier, and the user who executes the action. We propose the start and end of data listing, metadata, and data unit read operations as basic triggers since they enable tracking views and usages of different datasets at a data unit granularity. Additional statistics can be collected and associated with a query identifier, e.g., data transfer volume or consumed quota information from the provider.

Statistics collection is an essential prerequisite for offering multiple features. It enables reporting for data providers and users on which data is used and to what extent. Users can also be supported in adequately citing the data they use by automatically generating citations from the corresponding metadata, which leads to the provider receiving proper attribution for its valuable data. Additionally, statistics play a crucial role internally for caching by providing valuable information about how "cache-worthy" different data units are based on their past query performance and access profile. Statistics collection is also required for quota tracking at external providers as well as for internal quotas and/or billing purposes for users.

3.2 Authorization

Providing means for authorization is critical for enabling a multi-user environment. XCIP is not only about supporting the authorization of resources within the system where XCIP is implemented but also about synchronizing with the authorization of data providers so that we can ensure that their policies are enforced when accessing their data. This means that each access of a data provider's data needs to use the access control mechanism of the provider, e.g., by always using a user-provided token issued by the data provider. However, when sharing resources that involve data from a data provider, e.g., a pipeline that consists of loading data from a provider and further processing it, we also need to ensure the authorization means. In particular, access permissions of the original user are

not transferred to a different user that is interested in the resource. Thus, we want to keep the access permissions separate from the resource so that when another user accesses the resource, her/his authorization means are used. To enable this, we introduce the concept of a *provider configuration template* (PCT). A PCT consists of all the configurations necessary to connect to a provider except the authorization means. For example, a PCT for an Amazon S3 bucket could consist of the S3 endpoint URL and bucket name but miss an access token required to connect to the service. The authorization means, e.g., access tokens are kept separately for each user and inserted into a PCT only at query time to enable the connection to the provider with the permissions of the specific user running the query. This guarantees that each query to a data provider is only ever performed with the authorization means the querying user has provided, and no unwilling sharing of authorization means is possible.

3.3 Caching

Each step of the SDExL can benefit from caching, which prevents repeated and redundant queries to a provider. Operating a cache benefits the provider by reducing the load on its infrastructure. Each query that can be answered from the cache does not require a query at the provider. The only disadvantage for the provider is that since it does not receive cached queries, it cannot correctly track the usage of the data it provides. We address this issue with the statistics collection and reporting capabilities described above. The user benefits from caching because it decreases query response times, enables an interactive workflow, and reduces quota consumption at the provider.

Caching operates on data units such that the filter conditions of the data units control the granularity of the cache. Their identifiers are used to look up cache entries. Since they are associated with a filter condition, it is possible to answer queries with cache entries that do not exactly match a query by combining multiple entries and/or filtering them further. For example, consider the case where an existing cache entry's filter condition covers the query's filter condition. Then, the cache entry is sufficient to answer the query by applying the query's filter condition to the cached data. When the disjunction of multiple cache entries' filter conditions covers the query, their union can be used in the same way. If a query can be answered only partly from the cache, the provider must deliver the missing results. In case the provider supports filter pushdown, only the difference between the query and the cached data unit(s) must be retrieved from the provider. A cache replacement policy ensures optimal utilization of cache capacity. XCIP does not prescribe a specific policy, but insights gained from statistics collection should be used to weigh the importance of caching different data units. For example, keeping data units that require a long time to be retrieved from a provider in the cache should be favored over those that can be retrieved quickly. Suppose a user's quota at an external provider is exhausted but the user will likely reaccess a data unit before a new quota becomes available. In that case, it should also preferably be kept inside the cache. A challenge with caching is ensuring the freshness of cache entries when data changes over time. Ideally, a provider implements versioning such that updates to

data result in new external identifiers, and older versions stay accessible and unchanged under their old identifiers. If not, an expiry time should be attached to cache entries so they are periodically queried again at the provider, even if they would have otherwise stayed in the cache, to ensure their freshness.

Above, we described the separation between PCTs and authorization means for guaranteeing compliance with a provider's authorization policies. To be able to reuse cache entries across different users with different authorization means we need to integrate the authorization at query runtime into caching. We already established that sharing resources that use the data of providers is accomplished by only sharing PCTs and then connecting them with the querying user's authorization means. To make cache entries reusable between different users, we must ensure that data unit identifiers are equal for equal data regardless of which user's authorization means are used at query runtime. This is accomplished by making the provider identifiers, which are part of the data unit identifiers, refer to PCTs instead of fully configured providers. By referring to PCTs, which do not include user-specific configuration, the querying user does not impact the data unit identifier such that equal queries on equal data result in the same data unit identifiers. Since those queries then generate the same cache keys for all users, they can reuse the cache entries. It is, however, now necessary to check for the permissions of the querying user even when serving data from the cache. The reasons are that the permissions of the originally querying user either could have changed since cache insertion, or a different user could lack permissions altogether. Thus, we need to instantiate the associated PCTs and query for the permissions at the provider before returning a query response from the cache. In many cases, providers offer a metadata query that informs about users' permissions on a dataset without querying the actual dataset. Such queries are substantially faster than data queries and, thus, do not significantly impact the cache's query response time. In case a provider does not offer such metadata querying capabilities, actual data queries are necessary for determining the permissions such that caching becomes inapplicable without relaxations of the authorization, e.g., not rechecking a user's permissions once they were verified.

4 Case studies

We have implemented parts of XCIP in addition to SDExL within the spatiotemporal processing platform Geo Engine [Be23b]. Geo Engine offers powerful workflow-based processing of data from multiple sources. By implementing XCIP, we unify how data from different providers can be accessed within Geo Engine. This section presents a few examples of how XCIP applies to selected providers. They cover all parts of XCIP already implemented within Geo Engine and help illustrate how the unique properties and datasets offered by different providers can be handled.

Regardless of which provider's data is accessed, Geo Engine loads data through a source operator of a workflow operator graph that describes the data flow from data source(s) to workflow result. Between a source and result operator, various transformations can be

applied to the data, such as joining different datasets and computing functions over them. Since Geo Engine is a spatiotemporal processing platform, each operator's output is either of type vector or raster and has a spatial and temporal dimension. A user's query consists of a spatiotemporal filter condition. In addition to other filter conditions occurring in the operator graph on the path between the source operator and workflow result, this filter condition is pushed down to a source operator and forwarded to the data provider, if applicable. The pushdown filter conditions are used to define data units for caching and statistics collection. For workflows with a raster result, Geo Engine generates tile-wise subqueries along a fixed grid to increase the number of exactly matching data units between spatially shifted queries.

4.1 Copernicus Data Space

The Copernicus Data Space[7] offers data products through a SpatioTemporal Asset Catalogue (STAC)[8] service. Extensions of STAC exist, e.g., for raster data[9], which specify information about available raster bands. While the STAC service is used to discover and describe available data products, an S3 API is the primary access method for retrieving actual data. We take Sentinel-1 and Sentinel-2 satellite imagery (raster) datasets as examples of using an implementation of XCIP for STAC/S3 to integrate them into Geo Engine. The STAC service provides the required metadata for loading data from S3. It contains separate dataset entries for each timestep of Sentinel-1/Sentinel-2 data, while Geo Engine supports multiple timesteps within a single dataset. Therefore, a transformation is necessary to map multiple STAC entries to multiple timesteps in a single dataset in Geo Engine. The S3 API allows partial byte-range reads of objects, which we utilize to perform filter pushdown when applicable. For example, the Sentinel-1 data is available in a Cloud Optimized GeoTIFF file format[10], which contains a spatial index we can use to request only the relevant byte ranges of the remaining file. In contrast, Sentinel-2 data is only offered in JPEG 2000 format, which lacks an index and thus needs to be retrieved entirely before being able to filter irrelevant parts of the dataset. After a dataset or its relevant part has been retrieved, it is inserted into the cache, which allows for faster subsequent queries. Since Sentinel data is raster data that covers a large geographical region of which, according to our observation, often only a small contiguous region is required for a given query, we define data units as tiles dividing the world into smaller rectangles defined by their geographical extent. The size of the tiles is chosen so that most queries can be answered with only one or a few neighboring tiles. For an incoming query, we look up the required tiles in the cache, which then requests only the missing tiles from the S3 endpoint. In the case of Sentinel-1 data, the spatial filter condition can be pushed down by utilizing the index information available in the GeoTIFF file. Thus, the required data can be requested selectively for each missing tile. In the case of Sentinel-2 data, the whole dataset needs to be retrieved again once a

[7] https://dataspace.copernicus.eu/
[8] https://stacspec.org/
[9] https://github.com/stac-extensions/raster
[10] https://cogeo.org/

missing tile is encountered because of the lack of indexing information. However, keeping the most relevant parts of the Sentinel-2 dataset in the cache avoids cache misses, and the cache provides fast indexed access to the different tiles. The Copernicus Data Space applies quota limitations, which XCIP can monitor using its collected statistics, and caching helps to avoid running into rate limiting.

4.2 Aruna

Aruna [Di23a] is a storage orchestration engine that offers access to object storage. Data and its metadata can be accessed through an S3-compliant API or a custom API with advanced features, e.g., a relationship graph between objects, which Geo Engine uses since it models the connection between data and metadata objects for datasets. As Aruna stores arbitrary objects, we filter for custom Geo Engine metadata files tagged with a specific label. The relationship graph links each metadata object with one or multiple data files that constitute one dataset. Authorization plays an essential role in Aruna. Each object and higher-level construct, e.g., projects and collections, have associated (groups of) users who can be owners or have read permission. None of the other users have access to these resources. Aruna issues access tokens to authenticate the user in queries and check for sufficient permissions to read a resource. Geo Engine lets the user configure an access token for an Aruna connection. The token resembles the authorization means, while the remaining configuration serves as the PCT (e.g., the API endpoint). Aruna assigns a globally unique identifier to each resource, and a user either has at least read access to a whole object or no access to it. Thus, when all available metadata and data objects are mapped to datasets, XCIP can map to data unit identifiers consisting of the PCT identifier of an Aruna connection, a dataset's metadata object's resource identifier assigned by Aruna, and a filter condition. Using these identifiers, XCIP can apply statistics collection and caching. Authorization verification for cached data from Aruna is very efficient since its API supports a metadata query that returns a user's permissions on an object without querying the object itself. Geo Engine's cache uses an LRU (least recently used) cache replacement policy, which favors the most recently used cache entries to remain in the cache. Additionally, TTL (time-to-live) expiry ensures a minimum freshness of the cache. Aruna does not apply quota restrictions, so XCIP does not apply quota tracking but stores the collected statistics for data usage reports. From the metadata object associated with data objects through the relationship graph, XCIP generates a citation for the user.

5 Conclusions

This paper isolated several issues that arise when considering data integration based on partial materialization in the context of research infrastructures (RI). In particular, open issues are related to establishing a trustful relationship between an RI and its cooperating data providers to comply with the corresponding rules for authentication,

access policies, and quota restrictions. We proposed with XCIP (eXchange Component for Infrastructure Providers) the first initial approaches partly implemented in the context of NFDI4Biodiversity as an extension of our previous work that has already facilitated connectivity to data providers. In addition, XCIP offers certain performance benefits based on its powerful caching capabilities that combine a statistics-driven eviction strategy with filter pushdown for data providers.

Although our work focuses on spatiotemporal biodiversity data, our approach could be highly relevant for other settings and domains working with FAIR data. For example, our work could encourage other communities to rethink their current approach to data integration to build more trustful relationships between data users, data infrastructures, and data providers.

Acknowledgements

This work was partially funded by the German Research Foundation DFG under the grant agreement number 442032008 (NFDI4Biodiversity). The project is part of NFDI, the National Research Data Infrastructure Programme in Germany. This work was partially funded by the BMBF project FAIR Data Spaces (FAIRDS10). This work was partially funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant number 50EE2303B.

Bibliography

[BDF16] Buneman, Peter; Davidson, Susan; Frew, James: Why data citation is a computational problem. Communications of the ACM, 59(9):50–57, 2016.

[Be23a] Beilschmidt, Christian; Brandenstein, Dominik; Drönner, Johannes; Glombiewski, Nikolaus; Mattig, Michael; Seeger, Bernhard: On the Design and Implementation of Easy Access to External Spatiotemporal Datasets in NFDI. Proceedings of the Conference on Research Data Infrastructure, 1, Sep. 2023.

[Be23b] Beilschmidt, Christian; Drönner, Johannes; Mattig, Michael; Seeger, Bernhard: Geo Engine: Workflow-driven geospatial portals for data science. Datenbank-Spektrum, 23(3):167–175, 2023.

[DHI12] Doan, AnHai; Halevy, Alon; Ives, Zachary: Principles of data integration. Elsevier, 2012.

[Di23a] Dieckmann, Marius Alfred; Beyvers, Sebastian; Hochmuth, Jannis; Rehm, Anna; Förster, Frank; Goesmann, Alexander: The Aruna Object Storage: A Distributed Multi Cloud Object Storage System for Scientific Data Management. In: Proceedings of the Conference on Research Data Infrastructure. volume 1, 2023.

[Di23b] Diepenbroek, Michael; Kostadinov, Ivaylo; Seeger, Bernhard; Glöckner, Frank Oliver; Dieckmann, Marius Alfred; Goesmann, Alexander; Ebert, Barbara; Schimmler, Sonja; Sure-Vetter, York: Towards a Research Data Commons in the German National Research Data Infrastructure NFDI: Vision, Governance, Architecture. Proceedings of the Conference on Research Data Infrastructure, 1, Sep. 2023.

[Ji18] Jindal, Alekh; Karanasos, Konstantinos; Rao, Sriram; Patel, Hiren: Selecting subexpressions to materialize at datacenter scale. Proceedings of the VLDB Endowment, 11(7):800–812, 2018.

[Ka15] Karpathiotakis, Manos; Alagiannis, Ioannis; Heinis, Thomas; Branco, Miguel; Ailamaki, Anastasia: Just-In-Time Data Virtualization: Lightweight Data Management with ViDa. 2015.

[MB12] Mami, Imene; Bellahsene, Zohra: A survey of view selection methods. Acm Sigmod Record, 41(1):20–29, 2012.

[Ya21] Yang, Yifei; Youill, Matt; Woicik, Matthew; Liu, Yizhou; Yu, Xiangyao; Serafini, Marco; Aboulnaga, Ashraf; Stonebraker, Michael: Flexpushdowndb: Hybrid pushdown and caching in a cloud dbms. Proceedings of the VLDB Endowment, 14(11), 2021.

[ZBL23] Zhang, Mengxiao; Beltrán, Fernando; Liu, Jiamou: A survey of data pricing for data marketplaces. IEEE Transactions on Big Data, 9(4):1038–1056, 2023.

Keynote: Perspectives on Human Data Interaction

Laura Koesten [1,2,3]

Keywords: Human-Data Interaction, Data-centric Sensemaking, Data Visualization

Abstract:

In this talk, I will explore different perspectives on Human-Data Interaction – the study of how people engage with, experience, and make sense of data. My research aims to increase transparency in the decisions that shape data – from the questions we ask, the ways we collect and document data to how we analyze, communicate, and reuse data. A key challenge in this space is explaining both data and models to diverse audiences and developing tools that support these efforts.

I will present two related areas of research that investigate how people interact with data and models and how they understand them. The first area focuses on data-centric sensemaking and touches upon the concept of "data needs", and the notion of data as a dynamic process rather than a static artifact. The second area explores how different audiences interpret data through visualizations, with particular attention to the role of messages attached to visual data representations. Visualizations are everyday tools for communicating data, however, it is not clear whether audiences interpret them as their creators intended.

Drawing on my ongoing project, Talking Charts, I will illustrate these ideas with examples from research engaging both expert and general audiences across various visualization contexts. Through this, I aim to invite reflection on how we can design more meaningful data experiences.

Biography. Laura Koesten is an assistant professor at the Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), UAE, and affiliated with the University of Vienna, Austria. Her research focuses on improving human-data interaction by exploring sensemaking with data and visualizations, data discovery and reuse, as well as the ethical and collaborative aspects of data-centric work. She is the Principal Investigator of the WWTF Digital Humanism project *Talking Charts* and the recipient of the 2024 Hedy Lamarr Prize awarded by the City of Vienna, Austria. She earned a PhD in Computer Science from the University of Southampton, UK, in collaboration with the Open Data Institute, London, UK.

[1] University of Vienna, Faculty of Computer Science, Sensengasse 6, 1090 Vienna, Austria, laura.koesten@univie.ac.at, https://orcid.org/0000-0003-4110-1759
[2] Mohamed bin Zayed University of Artificial Intelligence, Department of Human-Computer Interaction, Masdar City, Abu Dhabi, UAE, laura.koesten@univie.ac.at, https://orcid.org/0000-0003-4110-1759
[3] AIT Austrian Institute of Technology GmbH, Center for Technology Experience, Giefinggasse 4, 1210 Vienna, Austria, laura.koesten@univie.ac.at, https://orcid.org/0000-0003-4110-1759

doi:10.18420/BTW2025-116

ODDA: Ontology-Driven Data Acquisition

Caspar Felix Hanika [2] [1], Dominik Sauerer [2 1], and Richard Lenz [1]

Abstract: Ontologies are powerful tools for structuring and formalizing knowledge and enabling interoperability. They play a crucial role in research data management by making data machine-interpretable and facilitating the integration of diverse datasets from various researchers. Ideally, the integrated datasets contribute to global use cases aimed at inferring new knowledge. To achieve this, it is essential that researchers align the data collected for their local use cases with existing domain or application ontologies. However, the data needs for local use cases might not readily fit into these reference ontologies without further adaptation. As a consequence, researchers often refrain from reusing existing ontologies and collect their local data on an ad hoc basis using improvised schemas.

We surveyed current approaches to ontology-based data-acquisition, revealing that while some methods exist for generating forms from ontologies, there are currently no significant approaches that enable researchers to customize these ontologies to fit their individual needs.

To address this gap, we developed a prototype aimed at exploring the challenges of adapting ontologies for specific use cases. Our form-based approach aims to empower researchers to reuse ontologies without requiring substantial expertise in ontology engineering. This prototype allows researchers to define ontology-based data-collection forms tailored to their specific needs, ultimately generating ontology-compliant knowledge graphs from the collected data.

Keywords: ontology-based knowledge acquisition, ontology, semantic web, form generator

1 Introduction

Research data, which are produced by different actors, are generally not comparable because of incompatible conceptualizations: For example, different terminologies are used for the same topics. Ontologies were defined to bridge this gap and establish a common vocabulary and semantic understanding. In theory, this should lead to better interoperability and interchangeability of research data [Sa24]. However, the practical usage of ontologies is niche at best, most use and reuse is seen in medical areas [SHH23]. This is exacerbated by the fact that often more specific tasks need more refined ontologies that might not be readily available. Unfortunately, this necessity of specific ontologies contradicts the global use cases and goals of ontologies. It would therefore be desirable to know the global use case in advance, i.e. global search and AI-readiness [SLH24], and enable domain experts to expand an existing application or domain ontology to a more specific ontology, that satisfies the researcher's needs.

[1] Friedrich-Alexander Universität Erlangen-Nürnberg , Department of Computer Science Chair 6, Martensstraße 3 91058 Erlangen, Germany, felix.hanika@fau.de, https://orcid.org/0009-0004-8963-7901; dominik.ds.sauerer@fau.de; richard.lenz@fau.de, https://orcid.org/0000-0003-1551-4824

[2] These authors contributed equally to this work.

doi:10.18420/BTW2025-117

A core issue is that domain experts who have the required knowledge about missing concepts or who want to store their research data using ontologies often refrain from using ontologies altogether or rely on extensive collaboration with ontology experts to do so. The main reason for this is that ontology languages like the Web Ontology Language (OWL) have a reputation for being too complicated and unapproachable [Tu20].
A main aspect of ontology research should be how to further enable domain experts, in our case researchers, to use ontologies and ontology-based data without the need for an ontology expert or extensive ontological experience. A common approach for ontology-based data acquisition are fillable forms created with form generators that use concepts of ontologies to describe the data. These forms resemble the schema the researchers have in mind for their data but use attributes and classes from a more general ontology (from here on called reference or target ontology) to describe the data. This also entails that form entries are stored using the same reference ontology concepts and thus the user's data is readily available for global use cases. Even within a single chemistry, lab integration issues may already arise making this concept desirable. One use for our envisioned tool is to enable single stations of the lab to specify their own data needs but facilitating integration by using the same reference ontology [SLH24].

The main goals of this paper are to investigate the current state of the art for ontology-based form generators, propose a prototypical form generator of our own, and discuss which issues remain. We are particularly interested in how person without extensive ontology knowledge can be enabled to use, extend, change, and access an ontology and ontology-based data.
The use cases we have in mind are researchers, i.e. for physics, biology or, chemistry, looking for opportunities to store their data based on ontological concepts and exchanging their data with one another.
The remaining work is structured as follows: chapter 2 presents the current status quo in research to ontology-based form generators, highlighting features about ontology extension and change. In chapter 3 we outline the main concepts of our prototype and in chapter 4 we evaluate our prototype in a first-experience way using an example use case based on the Wines & Foods ontologies created by the World Wide Web Consortium (W3C) [2]. In chapter 5 we propose further research topics based on our findings. Chapter 6 summarizes the most important findings of this paper.

[2] https://www.w3.org/2001/sw/WebOnt/guide-src/wine.rdf, https://www.w3.org/2001/sw/WebOnt/guide-src/food.rdf

2 Current state of ontology-based form generators

Before diving into the concrete results of our survey, we need to define clearly what we understand as an "Ontology-based/driven form generator": In this work, we define it as a tool that can produce forms manually or automatically, based on structures/concepts of any given ontology. These forms then can be used to insert data and populate the ontological concepts with individuals.
We provide a general overview of the work in this field and highlight specific tool features that support the extension of the underlying domain ontology.
For this survey, we used the following search engines: Google Scholar, DBLP, and SCOPUS advanced search. Our search terms were: "Ontology based data acquisition", "Ontology driven form generator", "Web-based Data Acquisition ontology", "Ontology population with web forms", "Ontology form acquisition", and "Ontology driven forms". From the results of these searches, we also included relevant works that were cited by our initial finding.
To narrow down the resulting paper collection, we only select papers that fulfil the following two criteria [Sa24]:

- Each paper must describe a tool, which represents an ontology-driven form generator. The main focus need to be on editing and filling forms rather than editing the ontology. This resembles more closely the more likely usage by a researcher and differentiates our work from a standard ontology-editor like Protégé [Tu13].
- RDFS and/or OWL ontologies must be supported, as these are the defacto standards in the context of ontologies.

It would also be desirable if the tool is not limited to a single specific ontology but supports importing an arbitrary ontology.
OBOP by Rutesic et al. [RRP21] lets the user import an OWL ontology, that the forms are based on. We will call this base or reference ontology from now on a "target ontology" and classes described in it as "target class" as the current prototype uses these terms, same for properties. To structure their forms OBOP uses its own ontology also called OBOP that links a form or its fields with target classes and properties. OBOP however does not provide functionalities to adapt the target ontology in any way and also does not assist the user in filling the form in a special way. Their focus is creating more complex form functions like conditional questions/form entries, i.e. only showing a form field if other fields have been filled with certain values.
Very similar is the work by Gonçalves et al. [Go17]: FACSIMILE also supports the import of different target OWL ontologies and uses a "data model" ontology to describe their forms, i.e. it defines a "Question". However, the structure of the form and linking between the form and target ontology is saved as an XML file. Notable is that the tool allows the inclusion of multiple target/domain ontologies as the mapping is described in the XML file. Like OBOP, FACSIMILE does not provide any features to expand the domain ontologies but allows conditional questioning.

Unlike the above OWL2MVC by Aydin and Aydin [AA20] does not support any complex form logic, but does propose a mechanism to help the form-filling process by collecting all possible database entries for a form field and providing them as options to the user.
Langer et al. [LGG21] propose CARDINAL. A key difference from other tools is that CARDINAL does not support multiple creatable forms. Instead, it has one basic form with a static part that remains constant and a dynamic part defined by the chosen ontology. However, the form part needs to be defined using the OnForm-Ontology and also this needs to happen outside the actual form generator, making this approach rather unsuitable for non-ontology experts.
Vcelak et al. propose to automate the form generation. Given an ontology, their tool creates forms for each class contained in the ontology. This produces rather simple forms, but they also consider additional restrictions like cardinality and multivalue constraints. Other features like ontology extension are missing.
The work by Sadki et al. [Sa18] operates similarly. Their tool also automatically generates forms relying on class properties and annotation properties. While the tool allows the editing of instances, no features are provided to change the underlying domain ontology.
Butt et al. [Sa13] provide a generator that allows for both automatic and manual form generation. They also include an entity lookup mechanism to help with form-filling. However, they do not offer any additional features that we would desire.
Another ontology-based form generator is PriSMHA by Goy et al. [Go20]. A main difference to the above is that their generator is designed specifically for the HERO ontology, containing concepts for describing historical data. PriSMHA is able to generate forms for specific use cases (in this case historic events) automatically. This is effective because, for different kinds of historic events, a survey was conducted, identifying the main properties and classes needed. However, the tool also does not allow any modifications to the ontology.

In conclusion, our investigation has shown that current form generators commonly lack features that help users define their own ontology concepts, adapt existing ones, or assist meaningfully in the form-filling process. Furthermore, questions regarding how a user can access their data or ensure data consistency after filling out forms remain unresolved by current work.

3 Concept for a ontology-based form generator

Besides regular functions to import ontologies, design, and fill forms our goals based on the results of chapter 2 are as follows:

- Including features to improve usability, aiming for better general user acceptance.
- Including features to specialize, extend, and check for consistency in ontologies and ontological data.
- Inclusion of these functions needs to be seamlessly integrated into the domain expert´s workflow.

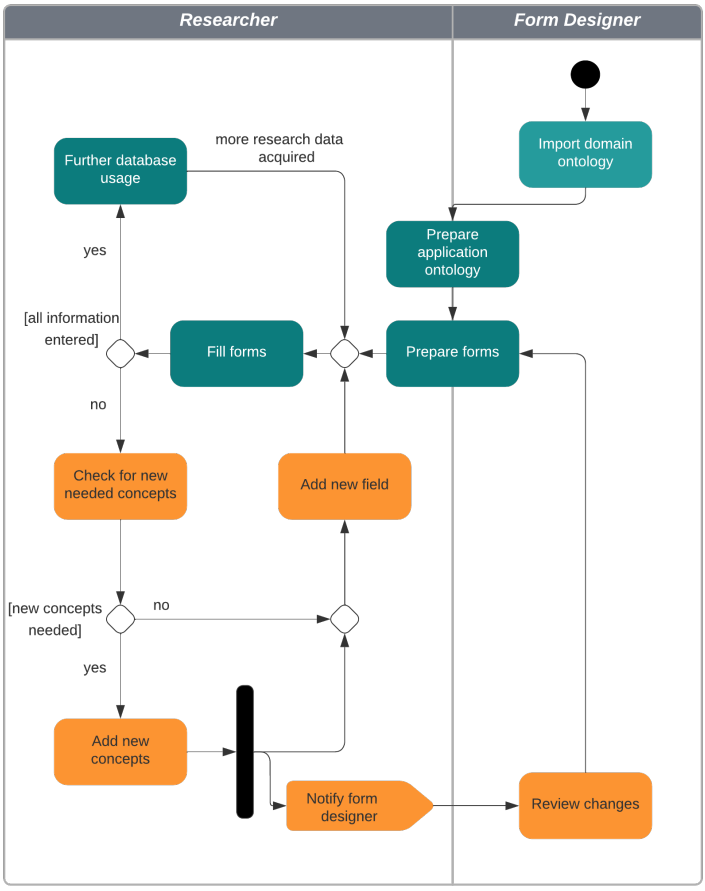


Fig. 1: Form generator workflow with ontology specialization elements

To address these requirements, we still assume that some basic expertise in ontology design is available when creating and adapting forms. However, we also aim to enable researchers with no ontology expertise at all to take full advantage of ontology-based data acquisition. For the sake of simplicity, we will differentiate between the roles of domain experts and ontology experts in the following sections. In a concrete use case, we expect the project participants to divide these roles based on individual expertise and background knowledge. Currently, we surmise the domain expert makes only punctual additions to the underlying domain ontology and forms, e.g. adding new fields to existing forms or adding another attribute to a class. Our long-term goal is to also enable the domain expert to add whole new classes to the target-ontology. Changes in the underlying ontology then still would have to be reevaluated by the ontology expert. The envisioned workflow is shown in figure 1.
To structure out forms we employ a dedicated form-ontology, describing different concepts like “FormElement”, “TargetClass” etc. The “TargetClass” i.e. specifying the class from the underlying domain ontology that the form is used to populate. Additionally, we introduce the notion of a draft: A draft is a not completely filled out form, which would not be allowed in the presence of any required fields. By allowing users to save drafts without adding them to the final knowledge graph, we enable them to pause their current work. This might be required in cases where the user realizes that other entries or additional form fields are needed first.
To further support the domain expert in the form-filling process we also use an entity collection mechanism similar to OWL2MVC that automatically collects suitable individuals in the knowledge base, based on the property´s range constraint and additional annotations, e.g. if the class name is included in property annotations. In the same way, we also support the form design process by proposing properties associated with the target class of the form [Sa24].
As stated before, one of our main goals is to enable domain experts to extend and specify the domain ontology according to personal needs. We do so by allowing the domain expert to add additional fields to forms. As in the initial design process of the form, the tool assists in identifying existing properties that could be added to the form. If none of the suggested properties suit the needs, we allow the user to create one. As we think of the user as a person without extensive ontology knowledge, we restrict the usage of OWL concepts to domain, range, and description. The user also chooses a name for the property, which the tool automatically extends to a valid IRI. In the background, this property also gets annotated with an “isUserDefined”-tag to distinguish between original ontology concepts and later added ones. Via this tag an ontology expert also can easily identify these user properties for later inspection [Sa24].
Similarly, we also allow ad-hoc additions of properties to individual instances, when filling out a form. This can be used in cases, where the form does not need to be changed in general because the property is only relevant for one or few entities.
A problem connected to the creation of new properties is the selection of range values in cases where the property object is an entity of another class. Since the amount of possible classes may be enormous, a simple drop-down list may not be suited. We instead use a graphical presentation of the ontology, showing the subclass relationships, which the user

can select a class from. In our opinion, a graphical interface better suits the task than a drop down as it makes it easier to identify the appropriate class by showing strongly related concepts for context. An example of this is shown in figure 2. Keep in mind that this example is used in the evaluation part of this work, but it shows the user interface in principle [Sa24].

Another issue that we want to address is to check the knowledge graph and the ontology for consistency after changes or insertions of new entities. The tool offers two ways to ensure consistency. Firstly, requirements and cardinality constraints like max/min-counts are checked directly when filling the form, in a similar fashion to Vcelak et al. [Vc17]. This approach could be used for other OWL restrictions too, but would require custom solutions for each constraint. Secondly we use a reasoner, in our case HermiT[3]. Currently, we check the knowledge graph for consistency with the reasoner after every change. This allows the tool to reject invalid data directly when inserted [Sa24].

4 Evaluation

In this chapter, we want to evaluate if the goal of assisting non-ontology experts in using and extending ontologies has been achieved by our prototype[4] in a practical way. Unfortunately, for this paper, a full usability study was out of scope and is an open topic that needs to be addressed in the future. Thus, we currently could only check if the features described in chapter 3 are working as intended, but cannot assess if they suffice in increasing usability for non-ontology experts besides observations of our own. Therefore, this evaluation is more of a first-experience report, but we hope this sparks some discussion about further research topics in the field.
Using the Foods and Wines ontologies as a base we could show that basic functionalities for generating, editing, and filling forms are working as planned. The more interesting topics need further investigation [Sa24].
As our contribution over the existing work, our form generator enables the user to extend ontologies and add additional properties to existing concepts. One of the main aspects was providing a minimized list of candidates when creating properties or filling out forms. In the case of form filling, the enhanced entity collection mechanism offers great help in reducing the possible choices for the user. However, our graphical approach for choosing candidates for range constraints, i.e. when creating new properties, quickly reaches its limits. In the case of the Foods and Wines ontologies, many classes are direct subclasses of owl:Thing (the superclass of all things in OWL) leading to an indistinguishable blob of classes as shown in figure 3. This suggests that current visualization techniques still have significant room for improvement.

[3] https://github.com/phillord/hermit-reasoner/tree/master
[4] https://github.com/Domi020/OntoFormGenerator

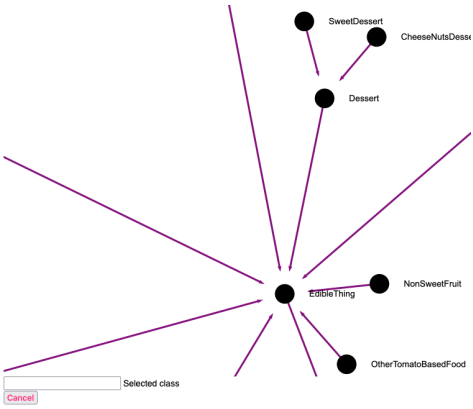


Fig. 2: Screenshot of the subclass graph (taken from [Sa24])

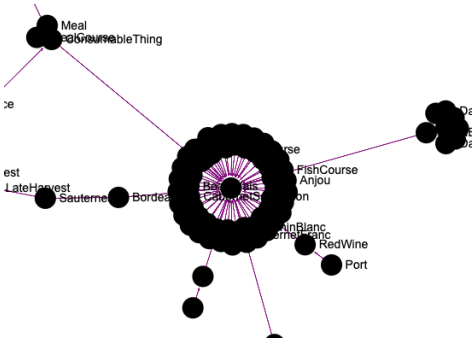


Fig. 3: Excerpt of a subclass graph with too many direct subclasses of owl:Thing (taken from [Sa24])

Another aspect when creating new properties is assisting the user in adhering to naming conventions or using meaningful names. Currently, we do not check the names entered by the user, allowing nonsensical names. The underlying philosophy is that domain experts should document all data according to their local use case needs and not be restricted by the imported domain ontology. At the same time, integrating this data into the domain ontology as far as possible aims to ensure compatibility for global use cases as far as possible. However, this may result in different researchers at various locations making incompatible additions for their specific local use cases, which is an inevitable consequence of uncoordinated decentralized customization. However, since the imported domain ontology is not modified but only extended, this should not lead to problems with global use cases that only rely on global definitions.

Our approach for using a reasoner for consistency checks also has significant drawbacks. The runtime behavior of the reasoner exhibits exponential growth in the size of the knowledge graph. This is shown by graph 4, which shows processing times on an Apple M2 processor with 16GB RAM for different generated knowledge graphs. We evaluated the runtimes of two reasoners HermiT and JFact[5] with consistent and inconsistent knowledge graphs adding additional information about wines indicated by a C (Chardonnays) or additional orders O that are the result of filling an order-form [Sa24]. The biggest problem is that the correct/consistent case is taking the longest time. This is of course antithetical to a general use case, indicating that the approach of constraining the initial form-filling process like Vcelak et al. may be preferred.

[5] https://jfact.sourceforge.net/

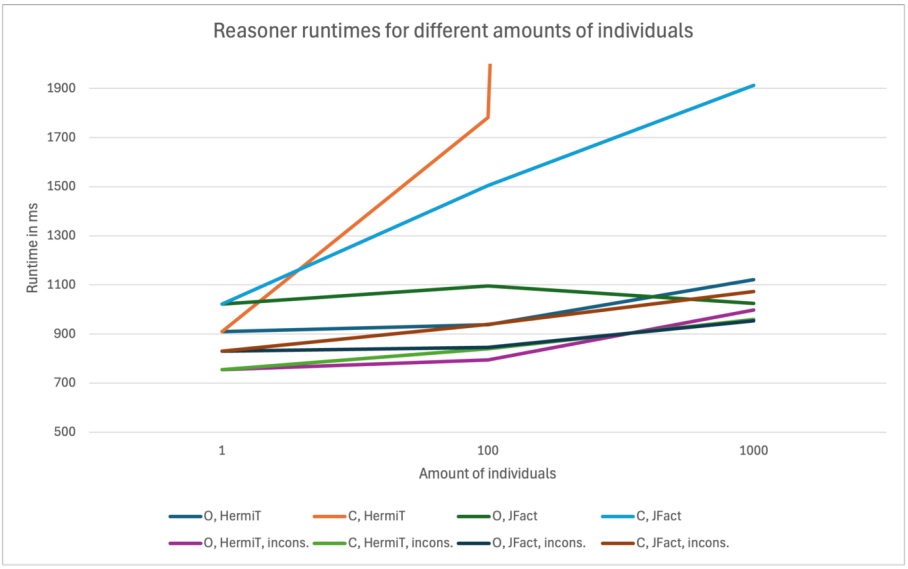


Fig. 4: Reasoner runtimes: Graphical depiction (taken from [Sa24])

An open issue that we also did not address yet, is the problem of data access. Of course, RDF data can be accessed via SPARQL, but it is very debatable if we can expect a person without ontology know-how to be able to use SPARQL. This needs further research that we indicate in chapter 5.

5 Open topics and future work

As discussed in chapter 4 there is still a lot of room for improving the domain experts experience using ontology-based data and ontologies themselves. One of the main problems is the difficulty in recognizing relevant ontology concepts or the lack thereof. In the context of a researcher aiming to store their research data in an ontology-based manner, a pre-mapping of the researcher's data schema to the appropriate ontology concepts would be desirable. In the same category falls the creation of forms in the first place: the researcher could further be assisted by being given a predefined pattern/form for a certain topic. For example, a form for storing data about bacteria could already propose form fields for regular fields other researchers normally use, like Gram-positivity.
Another remaining issue is the extension of ontologies by the domain experts themselves. While we already propose steps to assist this, we currently have no proper tools to guide where and when an ontology extension is needed. Furthermore, we cannot ensure interoperability if several domain experts extend the base domain ontology by themselves in parallel,

even if the same real-world object is being described. To resolve this, ontology/schema matching algorithms are needed to resolve integration issues between these modified ontologies.
Other concerns arise regarding the visualization of ontologies. As shown current ontology visualization has certain limits. More appropriate would be an approach that regards only relevant ontology concepts and hides the rest. This could be achieved in a similar way to the pruning of possible choices when filling the form as described in chapter 3.
As shown by figure 4 the current use of reasoners to validate the database and check for consistency is not sustainable for a growing knowledge base. We already discussed in chapter 4 that an approach to limit the possible entries while filling a form might be the better way. But reasoners would enable very complex checks and their usage might still be desirable. A possible solution could be to restrict the input of the reasoner by developing incremental checks or only check partial subgraphs of the knowledge base. This might mitigate runtime concerns but needs further research on how to decide what part of the graph would be needed.
An issue that we have not addressed at all yet is the way how non-ontology experts access their ontology-based data. In the space of RDF-based data storage, this would require knowledge of SPARQL, which we have to assume is nonexistent. For this reason, algorithms that either directly produce SPARQL-queries based on the form structure or produce the query based on natural language are needed, i.e. like current approaches with large language models [PB24; Ya23] or TEXT2RDF work [KH16; MAS21].

6 Conclusion

In this work, we investigate the current state of ontology-driven form generators. The survey revealed that such tools do exist but usually provide little support for extending an existing ontology to address individual needs. Based on these results we propose a form generator including features for ontology extension and assisting the general form-filling process, that current approaches are lacking. Our main goals are to enable domain experts without knowledge about ontologies to extend an existing ontology to suit their specific needs and store their data in a knowledge graph based on community-consented domain or applications ontologies. This would be preferable over creating an ontology from scratch or ignoring the technology altogether. We successfully implemented a prototype demonstrating essential features for an ontology-driven form generator using accessible technologies. The prototype addresses dedicated functionalities for ontology extension, allowing domain experts without ontology knowledge to generate ontology-based knowledge graphs. Although the evaluation is currently more like a first experience report, we already can identify multiple open challenges, including the need for real-world applications and user experience studies. We intend to further develop and evaluate this tool in the context of NFDI projects such as the Catalysis use case described by Stangl et al. [SLH24].

References

[AA20] Aydin, S.; Aydin, M. N.: Ontology-based data acquisition model development for agricultural open data platforms and implementation of OWL2MVC tool. Computers and Electronics in Agriculture 175, Publisher: Elsevier B.V., 2020, ISSN: 01681699, DOI: 10.1016/j.compag.2020.105589.

[Go17] Gonçalves, R. S.; Tu, S. W.; Nyulas, C. I.; Tierney, M. J.; Musen, M. A.: An ontology-driven tool for structured data acquisition using Web forms. Journal of Biomedical Semantics 8 (1), Publisher: BioMed Central Ltd., 2017, ISSN: 20411480, DOI: 10.1186/s13326-017-0133-1.

[Go20] Goy, A.; Colla, D.; Magro, D.; Goy, A.; Colla, D.; Magro, D.; Radicioni, D. P.; Accornero, C.; Loreto, F.; Radicioni, D. P.: Building Semantic Metadata for Historical Archives through an Ontology-driven User Interface. ACM Journal on Computing and Cultural Heritage 13 (3), 2020, DOI: 10.1145/3402440.

[KH16] Khilwani, N.; Harding, J. A.: Managing corporate memory on the semantic web. Journal of Intelligent Manufacturing 27, pp. 101–118, 2016.

[LGG21] Langer, A.; Göpfert, C.; Gaedke, M.: CARDINAL: Contextualized Adaptive Research Data Description INterface Applying LinkedData. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 12706, ISSN: 16113349, Springer, pp. 11–27, 2021, ISBN: 978-3-030-74295-9, DOI: 10.1007/978-3-030-74296-6_2.

[MAS21] Moteshakker Arani, Z.; Abdollahzadeh Barforoush, A.; Shirazi, H.: Representing unstructured text semantics for reasoning purpose. Journal of Intelligent Information Systems 56 (2), pp. 303–325, 2021.

[PB24] Perevalov, A.; Both, A.: Towards LLM-driven Natural Language Generation based on SPARQL Queries and RDF Knowledge Graphs. 2024.

[RRP21] Rutesic, P.; Radonjic-Simic, M.; Pfisterer, D.: An Enhanced Meta-model to Generate Web Forms for Ontology Population. In (Villazón-Terrazas, B.; Ortiz-Rodríguez, F.; Tiwari, S.; Goyal, A.; Jabbar, M. A., eds.): Knowledge Graphs and Semantic Web - Third Iberoamerican Conference and Second Indo-American Conference, KGSWC 2021, Kingsville, Texas, USA, November 22-24, 2021, Proceedings. Communications in Computer and Information Science, Springer, Kingsville, Texas, USA, pp. 109–124, 2021, DOI: 10.1007/978-3-030-91305-2_25.

[Sa13] Sahar Butt, A.; Haller, A.; Liu, S.; Xie, L.: ActiveRaUL: A Web form-based User Interface to Create and Maintain RDF data. In (Blomqvist, E.; Groza, T., eds.): Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013. Vol. 1035, CEUR-WS.org, Sydney, Australia, pp. 117–120, 2013.

[Sa18] Sadki, F.; Bouaud, J.; Guézennec, G.; Séroussi, B.: Semantically Structured Web Form and Data Storage: A Generic Ontology-Driven Approach Applied to Breast Cancer. In: Studies in Health Technology and Informatics. Vol. 255, ISSN: 18798365, IOS Press, pp. 205–209, 2018, ISBN: 978-1-61499-920-1, DOI: 10.3233/978-1-61499-921-8-205.

[Sa24] Sauerer, D.: A Form Generator for Ontology-Driven Data Acquisition, https://github.com/Domi020/OntoFormGenerator/blob/main/Masterthesis_OntoFormGenerator.pdf, 2024.

[SHH23] Shimizu, C.; Hammar, K.; Hitzler, P.: Modular ontology modeling. Semantic Web 14 (3), pp. 459–489, 2023.

[SLH24] Stangl, P.; Lenz, R.; Haumann, M.: The Use Case of Ontology-Driven Data Capturing for Machine Learning-Accelerated Catalyst and Reaction Design, Manuscript submitted for publication, 2024.

[Tu13] Tudorache, T.; Nyulas, C.; Noy, N. F.; Musen, M. A.: WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web. Semantic web 4 (1), pp. 89–99, 2013.

[Tu20] Tudorache, T.: Ontology engineering: Current state, challenges, and future directions. Semantic Web 11 (1), pp. 125–138, 2020.

[Vc17] Vcelak, P.; Kryl, M.; Kratochvil, M.; Kleckova, J.: Ontology-based Web Forms – Acquisition and Modification of Structured Data. In (Li, Q.; Wang, L.; Zhou, M.; Sun, L.; Qiu, S.; Liu, H., eds.): 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017). IEEE, Shanghai, China, pp. 1–5, 2017, ISBN: 978-1-5386-1937-7.

[Ya23] Yang, S.; Teng, M.; Dong, X.; Bo, F.: Llm-based sparql generation with selected schema from large scale knowledge base. In: China Conference on Knowledge Graph and Semantic Computing. Springer, pp. 304–316, 2023.

The Use Case of Ontology-Driven Data Acquisition for Machine Learning-Accelerated Catalyst and Reaction Design

Philipp Stangl [1], Richard Lenz [2], and Marco Haumann [3]

Abstract: The field of catalysis research is advancing through the application of machine learning. By predicting catalytic performance on the basis of experimental data, research time and costs can be reduced. However, inconsistent data formats and varied terminology complicate the merging of datasets to obtain a sufficiently large dataset for machine learning. With ontology-driven data capture, we aim to reduce ambiguity in the interpretation of experimental data, thereby enhancing the comparability of datasets for machine learning. We have found that combining a modular approach to modeling experimental data in an ontology with the reuse of existing ontologies allows for a structured way to capture local experimental setups while ensuring data compatibility beyond the local lab. This is demonstrated through the example of an ontology for capturing experimental data for industrially relevant selective oligomerization catalysis.

Keywords: Catalysis, Experimental Data, Machine Learning, Ontology, Research Data Management

1 Introduction

Catalysis research is entering a new phase characterized by the large-scale analysis of experimental data. One prominent approach in this field is the application of Machine Learning (ML), a subfield of data science that offers insights from data without requiring detailed knowledge of underlying kinetic, thermodynamic, or mechanistic principles. Instead, ML models rely on descriptors representing experimental settings and the resulting performances. ML models facilitate the development of predictive models that capture the relationships between experimental properties and catalytic performance. These models estimate outcomes, such as catalytic activity or desired selectivity, based on input parameters derived from past observations. According to their predictions, ML models can provide guidance for systematically selecting experimental settings, thus helping to reduce both research time and costs compared to conventional trial-and-error methods.

Despite the promising potential of ML in catalysis research, several challenges hinder its effective application. A major challenge is availability of comparable experimental data for large scale analysis. Often, research data necessary for training ML models is insufficient, incomplete, or missing altogether, limiting the ability of the model to generalize across

[1] Friedrich-Alexander-Universität (FAU), Lehrstuhl für Chemische Reaktionstechnik, Egerlandstraße 3, 91058 Erlangen-Nürnberg, Germany, philipp.stangl@fau.de, https://orcid.org/0009-0007-4179-2365
[2] Friedrich-Alexander-Universität (FAU), Lehrstuhl für Informatik 6 - Datenmanagement, 91058 Erlangen-Nürnberg, Germany, richard.lenz@fau.de, https://orcid.org/0000-0003-1551-4824
[3] Research Centre for Synthesis and Catalysis, Department of Chemistry, University of Johannesburg, P.O. Box 524, Auckland Park 2006, South Africa, marco.haumann@fau.de, https://orcid.org/0000-0002-3896-365X

doi:10.18420/BTW2025-118

different catalytic systems. Moreover, the lack of diverse and representative datasets can lead to biased models that fail to capture the full range of catalytic behaviors. In addition, the heterogeneous nature of experimental data—collected in different laboratories using various methods, conditions, and formats—requires significant preprocessing to ensure it is compatible with ML algorithms. These preprocessing steps, to standardize, clean, and integrate data from disparate sources, can be time-consuming and prone to errors.

Beyond catalysis, this issue of data scarcity is prevalent across various fields, including medicine and other scientific disciplines, where the application of ML methods often fails due to insufficient training data or too few case studies. To address these challenges, a method for combining diverse experiment datasets into a unified, comprehensive dataset could significantly expand the available training data. To ensure usability and comparability, the data must not only be collected but also documented systematically, capturing the right features consistently across different research locations. Discrepancies in recorded parameters, for instance, can severely limit the usability of the data for ML.

High-quality, well-annotated datasets are scarce in specialized fields like catalysis, where data can be expensive and time-consuming to generate. This work explores the application of ontologies to ensure a certain level of data homogeneity from the outset. It allows researchers to describe their data in such a way, that it becomes compatible with the data of other researchers for ML applications.

2 Background

First, we provide background on catalysis and reaction design in Section 2.1. Subsequently, we outline machine-learning assisted catalyst design in Section 2.2. Finally, we provide a brief overview of approaches for capturing experimental catalysis data in Section 2.3.

2.1 Catalysis and Reaction Design

Catalysis is a fundamental process in chemistry that accelerates chemical reactions through the use of catalysts, which are substances that increase the rate of a reaction without being consumed in the process, typically by lowering the activation energy barrier. Catalysts can be classified into two main categories: homogeneous catalysts, which exist in the same phase as the reactants, and heterogeneous catalysts, which are in a different phase. The design of catalysts is crucial for optimizing various chemical processes, including industrial applications and environmental remediation. Reaction design, on the other hand, refers to the strategic planning and execution of chemical experiments to achieve the desired performance. This involves understanding the kinetics and mechanisms of reactions, as well as the interactions between catalysts and reactants.

2.2 Machine Learning-Assisted Catalyst Design

Recent trends in catalysis and reaction design include computational modeling and ML. Computational modeling involves using advanced simulations to predict catalyst performance and reaction outcomes. Machine learning-assisted catalyst design is characterized by a significant evolution in methodologies, research data utilization, and the integration of advanced computational techniques. One of the primary advantages of ML in catalyst design is its ability to quantify the relationship between catalyst formulation and performance. For instance, supervised learning techniques have been employed to identify key synthesis variables that influence catalytic activity, thereby guiding the design process towards more promising candidates [Me24]. This approach not only enhances the efficiency of catalyst discovery but also allows for the identification of hidden trends within large datasets, which can inform future design strategies [Ta18; Ta19].

ML algorithms guide catalyst design and process optimization, for instance, based on key performance indicators (KPIs) [Ow24]. Fig. 1 illustrates the rate of a reaction r as a function of two parameters and its relation to the mechanism and theory. Mechanism describes how the chemical process takes place on a molecular level: How molecules behave in contact with the catalyst and how a reactant becomes a product. The mechanism of a reaction is often not known. Therefore, assumptions about the mechanisms can be made using key performance indicators (KPIs) derived from the reaction rate, which are then validated by experimental data. Alternatively, theory can make predictions about the reaction mechanism, or spectroscopic experiments can be conducted.

The field is also witnessing a growing emphasis on the development of comprehensive datasets that encompass a wide range of catalytic reactions and conditions. The Open Catalyst 2020 (OC20) dataset, which includes over a million density functional theory (DFT) relaxations, exemplifies efforts to provide a robust foundation for training ML models in catalysis [Ch21]. Such datasets are crucial for improving the generalizability of ML models across different catalytic systems and enhancing their predictive power.

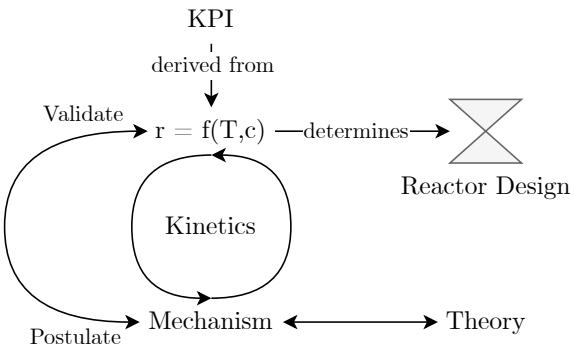


Fig. 1: Schematic representation of the domain.

2.3 Data Capture of Experimental Catalysis Data

The capture of experimental data in catalysis is a critical aspect in advancing this field. It ensures that the research findings are reproducible, accessible, and can be reused in future studies. A key strategy to improve data capture is the development of standardized protocols. Foppa et al. [Fo21] emphasize that standardized protocols enable the generation of consistent and annotated data. Furthermore, the integration of electronic lab notebooks (e. g., openBIS [Ba16]) and digital data repositories (e. g., Repo4Cat [KL24]) allows researchers to document and share experimental procedures, raw data, and metadata. However, these tools can often make standardization difficult, because they do not impose data-capture constraints to provide flexibility. In addition to tooling and standardized protocols, ontologies play an increasing role in enhancing data FAIRness [Wi16]. An ontology defines the entities that exist in a domain and the relationships between them [Gr93]. They use a shared vocabulary to describe a domain or subject area in a machine-understandable format and can facilitate the representation of complex catalysis data, thereby improving data interoperability [BBK24a]. Current ontology-based approaches focus on the development of a unified research data infrastructure [BBK24b]. However, no approach exists that uses ontologies to pre-harmonize experimental data in order to ease the data integration process for ML applications.

3 Use Case: Data-driven SCALMS Catalyst Design

Our use-case scenario for ontology-based data capture builds upon the foundational work of Owsienko [Ow24], which explored data-driven approaches for the design of Supported Catalytically Active Liquid Metal Solutions (SCALMS). SCALMS is a recently introduced concept of heterogeneous catalysis, in which a solid support is decorated with alloy particles. Under reaction conditions, these alloys liquify and the high dynamics in these liquids allows better catalysis (e. g., higher selectivity and lower deactivation) [Sø21]. The SCALMS catalysts consists of the building blocks support, matrix metal and active metal and the respective amount determines the performance. Hence, a large variety of SCALMS materials exist, which requires ML approaches to find the most suitable combinations.

3.1 Data Generation

Controlled laboratory experiments were conducted for SCALMS-catalyzed selective ethylene oligomerization (SEO) to study the influence of experimental variables on performance. Multiple experiments were independently performed by two researchers (A and B), who collected and stored their experimental data in Excel spreadsheets. The results of each experiment were either manually entered into spreadsheets or from raw text outputs of gas chromatographs, an analytical instrument that measures the content of various components in a sample. The dataset of researcher A comprises a total of 198 single-run experiments.

In parallel, researcher B studied SEO through 277 experiments that involved intentional variations in experimental parameters during a single run, generating a diverse dataset that reflects the impact of these varying conditions on the reaction outcomes. Additionally, as depicted in Fig. 3 (Appendix), the experiment context was captured in form of unstructured text, meaning that the extent of provided context varies significantly for each researcher.

3.2 Data Processing Pipeline

Following data collection, all data points from the various experiments were consolidated into an analytical base table (ABT). Each experiment was represented as a distinct data point, encompassing all relevant experimental parameters along with a metric that describes the corresponding catalytic performance. In the case of experiments, particularly those from the dataset by Researcher B, where parameters varied during the time on stream (TOS), the data was segmented to accurately map relationships between variables and outcomes. Each segment represents a period where the experimental conditions remained constant.

Differences in laboratory equipment and experimental setups among researchers resulted in incomplete documentation, that made the data difficult to compare and reuse. For example, Researchers A and B used slightly different catalyst compositions, reactor designs, and time intervals, which introduced variability and created gaps in data comparability and interpretation. Additionally, inconsistencies in temperature measurement units—Researcher A used Celsius while Researcher B reported in Kelvin—necessitated careful conversion and validation during the data integration process.

Additionally, there was no formal ontology or standardized vocabulary shared between the researchers, which led to inconsistencies in terminology and language usage. For example, column names in the dataset mixed languages with entries like “Ethan” (in German) and “Ethane” (in English), which complicated automated data processing and analysis. Moreover, different units of measurement were used for the same parameter, which required manual intervention and conversion and increased the potential for errors. Some entries also employed uncommon or non-standard terminology, such as “Ethan-1” (in German) instead of “Ethan” or “Ethane,” which created ambiguity that could obscure the identity of substances or experimental conditions.

Inconsistent protocols also resulted in missing information that is critical for the dataset to be useful for ML. Some experiments did not cover the full range of desired values for specific parameters (e. g., temperature, pressure). This led to a sparsity of data in certain ranges, making it difficult for ML models to make accurate predictions in these underrepresented areas. The data was made comparable through significant preprocessing efforts, which culminated in the output of an ABT. The preprocessing steps included mapping terms to a unified representation of terminology, outlier detection, and the calculation of KPIs (e. g., conversion, selectivity, deactivation rate). Tab. 1 provides an excerpt of the ABT.

Tab. 1: Analytical Base Table

Cat ID	Support	Ratio	TOS (h)	Reaction Temp. (°C)	Deactivation Rate (h^{-1})	Avg. Selectivity (%)
TN743	SiO2	49	50	359.85	-0.0153	87.56
AS036	SiO2	41	40	259.85	-0.0068	93.21
MM-15	AlOx	83	16	259.85	-0.0533	67.74
TN818	SiO2	49	80	199.85	0.0602	88.81
AS037	SiO2	50	63	239.85	-0.0035	91.77

3.3 Analysis

In a third step, the preprocessed data was analyzed through methods of visualization, statistical analysis, and predictive analysis. Data visualization techniques were applied to identify patterns, trends, and relationships. This step included the use of histograms for distribution analysis, radar diagrams for comparative analyses, and scatter plots to explore correlations between multiple dimensions. Statistical methods, such as correlation coefficients and dimensionality reduction (e. g., Principal component analysis), were used to assess relationships between variables. Clustering methods identified natural groupings within the data and helped to uncover distinct regimes or behaviors within the system. Predictive models were built using ML algorithms, with the dataset split into training and testing sets. Various models (e. g., Random Forest, Lasso, XGBoost) were used for predicting outcomes based on input parameters.

3.4 Validation

The models were evaluated using performance indicators such as the root mean squared error, with cross-validation to ensure generalizability. Feature importance and SHAP values were analyzed to understand the influence of variables on model predictions. The validated models were used to propose new experimental conditions aimed at improving performance. These new experiments provided feedback to further refine the models, creating a continuous improvement loop for optimizing system performance.

4 Rethinking Data Capture with Ontologies

Ontologies introduce a structured and machine-readable format, promoting the harmonization and standardization of experimental output. A domain ontology should serve as the foundation, providing a top-down structure that focuses on defining key concepts and relationships that are fundamental to catalysis. By capturing this type of knowledge in a structured ontology, the computer can understand catalysis-related terminology. It provides

a starting point upon which more specific and application-oriented ontologies can be built. Currently, a domain ontology for catalysis, dubbed Reac4Cat [BBK24a], is being developed collaboratively within the NFDI4Cat (National Research Data Infrastructure for Catalysis) [Es21] consortium. This collaborative effort involves multiple industrial and academic partners across Germany and ensures that the concepts and relationships in the domain ontology are widely accepted within the scientific community.

Researchers often have specific requirements and variations in how they conduct experiments, record performance data, and interpret results. To address the more local, context-dependent data capture needs in laboratories, application-specific ontologies can be used. These ontologies should be developed using a bottom-up approach, to ensure they are closely aligned with actual data requirements and aspects of capturing and harmonizing experimental data (e.g., comparable performance indicators) in real-world research settings.

One way to develop such application-specific ontology for catalysis experiments is through the Modular Ontology Modeling (MOMo) [SHH23] workflow. This approach builds on the eXtreme Design methodology but introduces the concept of modularity and the use of graphical schema diagrams. The workflow prioritizes breaking down a complex idea, typically organized in a monolithic ontology, into intuitive, self-contained modules, each capturing key concepts and relationships. Furthermore, a set of modules can be extended to accommodate new data that was not yet available during the data capturing, but may become available later. For instance, performance indicators are calculated in the data processing step (Sec. 3.2) and are not directly captured.

To realize the full potential of ontology-based data capture, ontologies must become integral components of the research workflow. Tools such as ontology-based form generators to design data entry forms based on the ontology allow researchers to input their experimental data in a structured way [HSL25]. The classes, properties, and relationships defined in the imported ontology will guide data entry. The ontology-annotated datasets can be exported as knowledge graphs, making the data readily accessible and usable for advanced analysis techniques, such as ML and automated reasoning.

5 SCALMS Ontology by Example

This section illustrates how an application-specific ontology for the SCALMS use case can be approached using the MOMo workflow based on the ABT. The first step is to define specific questions researchers want to answer using ML models. These questions will guide the ontology development process. For example:

- How does the atomic ratio of matrix and active metal (here: Gallium and Nickel) influence the product selectivity at different reaction temperatures?
- Does the support material impact the catalyst deactivation rate?

Based on the competency questions and the data schema, the core concepts that need representation in the ontology can be identified. These could include:

Catalyst : `Cat ID`, `Galium Loading`, `Nickel Loading`, and `Support`.

Reaction : `Temperature`, `Pressure`, gas flows for different gases (ethylene and helium in your example), and `Residence Time`.

Performance : `Conversion`, `Deactivation`, `Selectivity`, and `Product Yield`, each with initial, final, and average values.

A fundamental step before module development is to leverage existing ODP libraries (e. g., MODL [SHH19]) that can offer templates to represent key notions. MODL, for instance, defines the *Quantity* pattern, which can be used to model various numerical parameters like `Bed Loading`, `Galium Loading`, or `Reaction Temperature`. Next, it is important to develop graphical schema diagrams for each module that allows domain experts to grasp what each module represents and what it contains. This step is especially important, as domain experts (that is, researchers) must be able to apply the ontology. Fig. 2 illustrates the graphical schema diagram of the SCALMS use case ontology. It shows how concepts are bundled into modules and how they connect to each other. Modules have different purposes and roles within the ontology. A module like Reaction or Catalyst represents domain-specific concepts, focusing on entities and their functional or physical properties in catalysis.

Catalyst : Has properties for loadings, ratio, and support material. This module might also connect to other modules like *Material* to represent the chemical composition of the catalyst and support.

Reaction : With properties for temperature, pressure, gas flows, etc. It can connect to the *Catalyst* module to indicate the catalyst used in the reaction and to *Quantity* modules to represent the numerical values of reaction parameters.

Performance : Classes for *Conversion*, *Selectivity*, and *Product Yield* each with initial, final, and average values represented as data properties. This module can be linked to the *Reaction* module to associate performance data with specific reactions.

In contrast, other modules like Identifier and Quantity are domain-agnostic utility modules that provide a framework for handling data representation and abstraction. These modules are not tied to any specific field but are designed for reuse across various domains. The Identifier module, for instance, focuses on encoding and standardizing identifiers for entities like catalysts or reactions. Similarly, the Quantity module deals with representing measurable data, such as numerical values and their units (e. g., temperature in degree Celsius).

With the modules in place, axioms can be defined to govern the relationships between concepts. For instance, a *Catalyst* must have a *Cat ID*, *Support*, and at least one of *Galium Loading* or *Nickel Loading*. Similarly, each *Reaction* must be associated with a *Catalyst* and have Reaction Temperature and Pressure values. Finally, an OWL file can be generated to capture the structure and relationships of the modules.

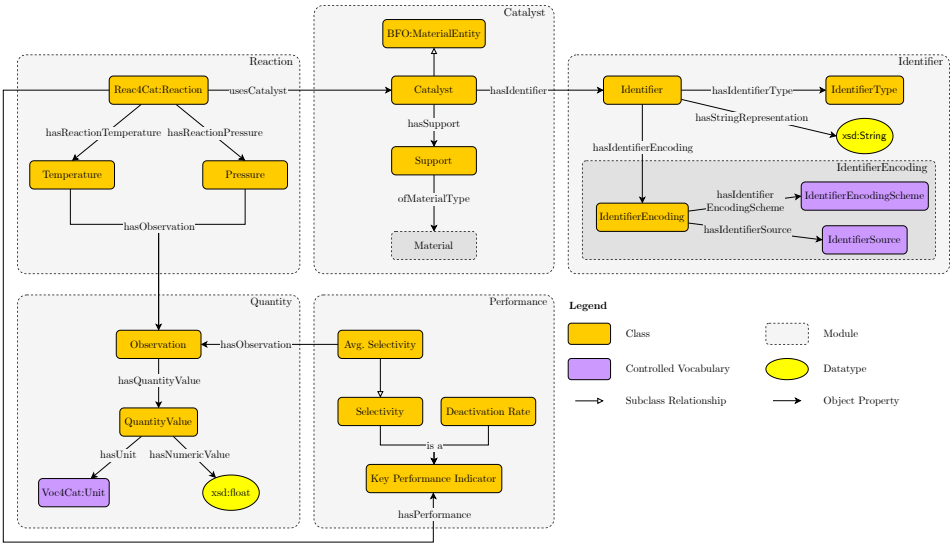


Fig. 2: Ontology schema diagram for the SCALMS use case.

Preliminary results show significant potential of ontology-driven data acquisition for experimental data. For example, by applying ontology-based mappings, we successfully reconciled key variables such as "reaction temperature" across the datasets, even when different units and terminologies were used. The ontology helped to automatically standardized these values, applying predefined conversion rules and aligning data to a unified schema. This process substantially reduced the manual data preprocessing effort, allowing us to consolidate disparate experimental datasets from multiple researchers into a consistent ABT at a faster pace than manually processing the data.

6 Planned Evaluation

The evaluation of the ontology will be carried out in two steps to ensure both the correctness of the ontology and its practical utility for the intended use case. First, ontology validation verifies whether the ontology is syntactically and semantically accurate and if it meets its intended purpose. Syntactic validation ensures that the ontology adheres to the rules of its specification language (e. g., OWL, RDF Schema) using syntax-checking tools. Semantic validation confirms that the ontology represents its domain correctly through several checks: detecting empty classes through a reasoner (e. g., HermiT [Gl14]), identifying contradictions using the OntoClean [GW09] methodology and validating the knowledge with domain experts. In addition, competency questions are used to ensure that the ontology meets predefined requirements.

Secondly, the practical application and usability of the ontology must be evaluated. Researchers will be surveyed to assess the ease of using the ontology-based system versus manual methods. The time required to annotate and curate data with and without the ontology will be measured to determine efficiency. Further, ontology-described data from multiple experiments will be merged to a single large dataset for ML. These results will be compared with previous research results achieved by Owsienko [Ow24] to assess whether the ontology improves performance or provides new insights.

7 Summary and Outlook

The readiness of research data in catalysis for ML applications hinges on the standardization and harmonization of data collection practices. To explore this, we analyzed the use case of leveraging ML-accelerated catalyst design for SCALMS within the context of research data management. Our study focused on integrating data from various researchers to create a large dataset for training ML models. This involved consolidating experimental data, often presented in diverse formats and terminologies, into a unified analytical base table. During this process, several challenges were uncovered. These included inconsistencies in temperature units, language barriers, and varying levels of experimental context. Such discrepancies arose from differences in laboratory setups or documentation practices and necessitate extensive manual efforts to make the data suitable for ML applications.

To address these issues, we propose to pre-harmonize experimental data with the use of ontologies. By adopting a modular methodology, we emphasize breaking down complex data structures into smaller, self-contained units, or modules. Each module is designed to represent specific aspects of the data, such as catalyst properties, reaction conditions, and performance metrics. These modules are interconnected to create a cohesive yet flexible structure that can be easily adapted to different experimental setups and research requirements. In addition to modularity, we reuse existing ontologies to promote standardization and interoperability beyond the local lab. Ontologies, such as those developed within the NFDI4Cat initiative, provide a foundational, widely accepted vocabulary and structure for describing key concepts and relationships in catalysis research. The manual effort required for integration and harmonization processes is significantly reduced through a standardized framework that aligns diverse datasets for our use case.

In future work, we aim to apply our approach to other reaction experiments and plan to integrate our methodology within the broader tooling landscape of NFDI4Cat. A significant focus of our ongoing efforts is the development of tools to create ontology-based forms by enabling the import of existing ontologies and the export of ontology-annotated data as knowledge graphs. Furthermore, we are addressing challenges such as managing ontology versioning. For projects involving normed entities, the integration of an established authority file system (e.g., GND) could further enhance data consistency and linkage with external reference sources. These efforts will enhance the robustness and scalability of our approach across diverse applications in the field of catalysis.

References

[Ba16] Barillari, C.; Ottoz, D. S.; Fuentes-Serna, J. M.; Ramakrishnan, C.; Rinn, B.; Rudolf, F.: openBIS ELN-LIMS: an open-source database for academic laboratories. Bioinformatics 32 (4), pp. 638–640, 2016.

[BBK24a] Behr, A. S.; Borgelt, H.; Kockmann, N.: Reac4Cat-Ontology: Harnessing the Power of Ontological Description Logic in Catalysis Research as a Practical Approach to Knowledge Inferences. Datenbank-Spektrum, pp. 1–12, 2024.

[BBK24b] Behr, A. S.; Borgelt, H.; Kockmann, N.: Ontologies4Cat: investigating the landscape of ontologies for catalysis research data management. Journal of Cheminformatics 16 (1), p. 16, 2024, DOI: 10.1186/s13321-024-00807-2, URL: https://doi.org/10.1186/s13321-024-00807-2.

[Ch21] Chanussot, L.; Das, A.; Goyal, S.; Lavril, T.; Shuaibi, M.; Riviere, M.; Tran, K.; Heras-Domingo, J.; Ho, C.; Hu, W., et al.: Open catalyst 2020 (OC20) dataset and community challenges. Acs Catalysis 11 (10), pp. 6059–6072, 2021.

[Es21] Espinoza, S.; Linke, D.; Wulf, C.; Schimmler, S.; Schunk, S. A.; Benner, P.; Kraehnert, R.; Kragl, U.; Palkovits, R.; Palkovits, S., et al.: NFDI for Catalysis-Related Sciences: NFDI4Cat. Bausteine Forschungsdatenmanagement 2, pp. 57–71, 2021.

[Fo21] Foppa, L.; Ghiringhelli, L. M.; Girgsdies, F.; Hashagen, M.; Kube, P.; Hävecker, M.; Carey, S. J.; Tarasov, A.; Kraus, P.; Rosowski, F., et al.: Materials genes of heterogeneous catalysis from clean experiments and artificial intelligence. MRS bulletin, pp. 1–11, 2021.

[Gl14] Glimm, B.; Horrocks, I.; Motik, B.; Stoilos, G.; Wang, Z.: HermiT: an OWL 2 reasoner. Journal of automated reasoning 53, pp. 245–269, 2014.

[Gr93] Gruber, T. R.: A translation approach to portable ontology specifications. Knowledge acquisition 5 (2), pp. 199–220, 1993.

[GW09] Guarino, N.; Welty, C. A.: An overview of OntoClean. Handbook on ontologies, pp. 201–220, 2009.

[HSL25] Hanika, F.; Sauerer, D.; Lenz, R.: ODDA: Ontology-driven Data Acquistion. In: Proc of the 5th Workshop on Big (and Small) Data in Science and Humanities (BigDS) in conjunction with the 21st Conference on Database Systems for Business, Technology and Web (BTW'25). Bamberg, DE, 2025.

[KL24] Kushnarenko, V.; Linke, D.: Repo4Cat - A Repository for Catalysis Data, Accessed: 2024-12-09, 2024, URL: https://github.com/nfdi4cat/repo4cat.

[Me24] Mendes, P.; Vermeire, F.; Van Haute, T.; Thybaut, J. W.: Machine learning for catalyst design: data matters. 2024.

[Ow24] Owsienko, A.: Statistical and predictive analysis of supported catalytically active liquid metal solutions (SCALMS), Master's thesis, Erlangen, Germany: Friedrich-Alexander-Universität Erlangen-Nürnberg, 2024.

[SHH19] Shimizu, C.; Hirt, Q.; Hitzler, P.: MODL: a modular ontology design library. arXiv preprint arXiv:1904.05405, 2019.

[SHH23] Shimizu, C.; Hammar, K.; Hitzler, P.: Modular ontology modeling. Semantic Web 14 (3), pp. 459–489, 2023.

[Sø21] Søgaard, A.; de Oliveira, A. L.; Taccardi, N.; Haumann, M.; Wasserscheid, P.: Ga–Ni supported catalytically active liquid metal solutions (SCALMS) for selective ethylene oligomerization. Catalysis science & technology 11 (23), pp. 7535–7539, 2021.

[Ta18] Takahashi, K.; Miyazato, I.; Nishimura, S.; Ohyama, J.: Unveiling hidden catalysts for the oxidative coupling of methane based on combining machine learning with literature data. ChemCatChem 10 (15), pp. 3223–3228, 2018.

[Ta19] Takahashi, K.; Takahashi, L.; Miyazato, I.; Fujima, J.; Tanaka, Y.; Uno, T.; Satoh, H.; Ohno, K.; Nishida, M.; Hirai, K., et al.: The rise of catalyst informatics: towards catalyst genomics. ChemCatChem 11 (4), pp. 1146–1152, 2019.

[Wi16] Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific data 3 (1), pp. 1–9, 2016.

8 Appendix

RUN ID	2022-07-11_SCALMS_JD_DIMER_86_TN729_350C_4x16h+oxidation		
DATE	22.07.2011		
OPERATOR	John Doe		
CATALYST ID	TN729		
CATALYST DESCRIPTION		Ga43Ni on SiO2	
CATALYST AMOUNT	1	g	
TEMPERATURE	350	°C	
PRESSURE	5	bar(g)	
RUN TIME	16	h	x 4
GAS FLOW	10	mL_N/min	C_2H_4 (Ethylene)
	40	mL_N/min	He (Helium)
PRETREATMENT	20	mLN/min	H_2, for 2h, 310°C, atm
EXPERIMENT COMMENTS			

(a) Researcher A

RUN ID	SCALMS-MM-MM17-DIMER-Ethen-16h			
DATE	28.02.2023	24.02.2023		
OPERATOR	Max Mustermann			
CATALYST ID	MM-17			
CATALYST DESCRIPTION			$Ga_{106}Ni$	
CATALYST AMOUNT	1,0056	g		
TEMPERATURE	260	°C		255
PRESSURE	5	bar(g)		5,2
RUN TIME	16	h		
GAS FLOW	10	mL_N/min	C2H4	
	40	mL_N/min	He (Helium)	
	2,98	L* gcat-1 * h-1		
PRETREATMENT	20	mLN/min	H_2, for 2h, 310°C, atm	
	1,19	L* gcat-1 * h-1		
EXPERIMENT COMMENTS	AlOx als Support			

(b) Researcher B

Fig. 3: Catalyst information provided by researchers for their datasets.

An overview of current ontologies for interdisciplinary ecosystem, biodiversity and agricultural research

Philipp Plamper[1], Thorsten Hauke[1], and Anika Groß[1]

Abstract: Global change negatively impacts ecosystems, biodiversity and agriculture. Intensive research efforts with numerous case studies in these fields generate large amounts of datasets that should be combined and comprehensively analyzed to synthesize new overarching findings. This requires very good research data management practices according to the FAIR data principles. Interdisciplinary projects should carefully use ontologies for semantic annotation to reduce terminology and data integration problems. Before developing new domain or application specific ontologies, suitable existing ontologies for reuse must be identified. This is a challenging task due to the ever-growing and complex landscape of new ontologies as well as various domain-specific ontology portals that should but may not simplify ontology search. Here, we provide an overview of ontology portals and ontologies, in interdisciplinary ecosystem, biodiversity and agricultural research, and discuss open challenges for ontology search and development.

Keywords: Ontology, FAIR, Ecosystems, Biodiversity, Agriculture

1 Introduction

Ecosystem functionality and biodiversity are increasingly compromised by global changes such as climate change and intensive land use [Ti16, Ma15], which negatively impact crucial *ecosystem services*, including drinking water, natural materials and food production [Mea05, BVH18, Po10]. Investigating ecosystem functionality and biodiversity in agricultural landscapes requires inter- and transdisciplinary research to integrate knowledge from different fields for a holistic understanding that helps to preserve ecosystem services for future generations. As with other application domains, this involves the collection and integration of various datasets, which are often created and managed autonomously resulting in heterogeneity. Heterogeneous data limits interoperability and hinders the ability to analyze data, reducing the added value and the gain in new insights.
Therefore, it is crucial that interdisciplinary projects meet the FAIR guiding principles [Wi16] adopting good data management and stewardship to enable and simplify data reusability, e.g. for comprehensive analysis. An important part is the consistent definition and usage of terminology within and across domains as well as the proper semantic annotation of data objects. To ensure a consistent understanding, ontologies aim to standardize and structure knowledge based on an explicit formal specification of a conceptualisation [Gr93].

[1] Hochschule Anhalt University of Applied Sciences, Department Computer Science and Languages, Lohmannstraße 23, 06366 Köthen, Germany,
philipp.plamper@hs-anhalt.de, https://orcid.org/0009-0000-9663-7361; thorsten.hauke@mailbox.org;
anika.gross@hs-anhalt.de, https://orcid.org/0000-0002-2684-8427

doi:10.18420/BTW2025-119

They provide a formal representation of knowledge by defining concepts (or classes) interconnected via semantic relationships. Despite the goal of standardization, there are several different formats for storing and exchanging ontologies, such as RDF/XML, RDF/TTL, OWL, SKOS and OBO and there are numerous domain ontologies with different but partly overlapping contents. The introduction of the "Biodiversity and Ecology (biodiv)" track by the Ontology Alignment Evaluation Initiative (OAEI) since 2018 [Ka20] emphasises the high demand for interoperability in these domains. The need for search and overview on existing ontologies, led to the development of various ontology portals, such as BioPortal [No09], EcoPortal [Ke21] or BiodivPortal [Ka24]. These portals collect, curate and provide ontologies, often with a focus on a specific domain. However, not all ontologies relevant to a domain or application are available in a centralized portal and this is even more the case for interdisciplinary integration.

Complex ecological projects situated at the interface of ecosystems, biodiversity and agriculture require careful and comprehensive research data management (RDM) including the use of ontologies for enhanced integration. For instance, in the "AgriRestore" project researchers from ecology, agriculture and data science aim to develop an in-depth understanding of key indicators for the transition to more resilient ecosystems and landscapes, based on elaborate experiments on restoration measures in agricultural landscapes (e.g. flower strips along fields) as well as comprehensive data analyses for evidence-based restoration [Fi24]. In this research process, standardized concepts and vocabularies will be needed to build comprehensive knowledge graphs as the backbone for data integration and analysis.

Fortunately, in the fields of agriculture, biodiversity and ecosystems, there are already various ontologies focusing on both, generic and specific aspects. Before developing new ontologies, it is crucial to have a thorough overview of the existing landscape of ontologies that are related to the considered datasets and research questions in order to exploit the full reuse potential. However, it is challenging to maintain an overview and identify suitable ontologies for a specific topic, because the ontologies of interest may be distributed across different ontology portals. We therefore outline current ontologies and relevant ontology portals for interdisciplinary research:

- We give a brief introduction to ontologies and their formats (Sec. 2) and show which different but related ontology portals are available (Sec. 3).
- We summarize our ontology search in the interdisciplinary ecosystem, biodiversity and agriculture context and provide an overview of 41 ontologies based on different categories such as available formats, FAIR score, year of the last update, visits, number of classes/individuals/properties and availability in different portals (Sec. 4).
- We discuss open challenges (Sec. 5) for ontology search and development in this interdisciplinary research field and conclude our findings (Sec. 6).

2 Ontology Formats

Ontologies are created and used to ensure consistent terminology within and between disciplines. They can cover a broad range of complexity and expressiveness, from basic controlled vocabularies and thesauri to both informal and formal hierarchical structures as well as formal ontologies that specify disjoint classes, part-whole relationships and other kinds of logical constraints [LM01]. Furthermore, they should be accessible for reuse by other actors in the data context [Ca20, Hi21] and should also be shareable with other interested parties in the respective domain [SBF98].
These requirements are in line with the FAIR data principles (Findable, Accessible, Interoperable and Reusable), which were formulated to improve the infrastructure for the reuse of scientific data [Wi16]. Ontologies serve as a common understanding of concepts and enable the uniform modeling and integration of knowledge. They are a central component of the vision of the semantic web [Hi21, BLHL23] where information is represented by common identifiers (Unique Resource Identifier, URI) and enriched with machine-understandable metadata to improve interoperability and data linking [BLHL23]. The linked graphs can be viewed as one large graph [Hi21]. Knowledge graphs [Ji22] are used to model real-world phenomena, especially when data is highly interconnected [Sa21]. A recent survey on knowledge graph construction [Ho24] defines a knowledge graph as an integrated graph that contains data from different sources, where the semantic structure of the knowledge graph is defined based on ontologies from one or more domains including the types and properties of entities and their relationships.
Ontologies are serialized for storage in different formats. The formats are used to describe, integrate and query semantic data as well as for structuring and standardization. Not every format is supported equally and available formats differ which impairs interoperability. One of the most common serialization formats is RDF/XML, which builds on top of the Resource Description Framework (RDF) for the data model and utilizes XML (Extensible Markup Language) for the syntax [BM04]. XML is widely used, so RDF/XML has the advantage of being interpretable by a large number of XML parsers. A commonly used notation for the creation and exchange of RDF-based knowledge is the Terse RDF Triple Language (Turtle, TTL), as it is more human-friendly and allows direct conversion to RDF [Co14]. Another widely used format is JSON-LD (JSON for Linked Data) [LG12].
The Web Ontology Language (OWL) is a W3C standard format and is based on description logic [Co12]. OWL provides a comprehensive language for defining ontological knowledge and enables complex models and conclusions to be represented in a formal and machine-understandable form. In addition, OWL has several syntaxes, e.g. Manchester syntax. It supports the creation of ontological axioms, classes, properties and individual instances. For instance, OWL is used in knowledge-intensive applications such as the semantic web and biomedicine [Hi08].
The Open Biological and Biomedical Ontologies (OBO) format is specifically tailored to the requirements of biology and biomedicine. OBO uses a syntax based on key-value pairs and enables the semantic modeling of biological classes, relationships and taxonomies while remaining human-readable [Ti11].

The Simple Knowledge Organization System (SKOS) [MB09] is a vocabulary based on RDF and specializes in the modeling of knowledge organization systems and taxonomies. It offers a collection of classes and relations for the representation of class hierarchies, thesauri and related structures. SKOS semantics enable simple encoding of classes, synonyms, associations and hierarchies, structuring and semantically enriching knowledge in a standardized, interoperable way.
Ontologies aim to standardize terminologies and create a common understanding to support semantic interoperability and reuse. However, there are a growing number of domain ontologies with related content, and it is difficult to keep track of them all or to choose the right one for a specific use case. This has led to the development of ontology portals designed to facilitate the discovery and reuse of ontologies.

3 Ontology Portals

An ontology portal provides a centralized access point that simplifies the discovery, evaluation and selection of ontologies from a wide range of relevant top-level to domain ontologies. With the increasing demand for ontologies from the fields of ecology and agriculture, several portals are dedicated to these domains and collect ontologies. We searched for relevant ontology portals to find suitable ontologies for future knowledge graph development a.o. in the context of resilient ecosystems, biodiversity and ecosystem and landscape restoration measures in agricultural landscapes.
The following portals were examined for the overview: BioPortal, AgroPortal, EcoPortal, Ontobee, Open Biological and Biomedical Ontology (OBO) Foundry, AberOWL, Ontology Lookup Service (OLS), Bioregistry, FAIRsharing, BiodivPortal and EarthPortal, which will be briefly introduced below.
Table 1 summarizes the ontology repositories with their corresponding URL, the number of currently available ontologies and whether or not a FAIR score is provided. The FAIR score is based on the O'FAIRe (Ontology FAIRness Evaluator) method and is used to check compliance with the FAIR principles of an ontology [AJ22, ABJ22]. The table shows that the total number of available ontologies is larger on portals that focus more on biology and biomedicine than on portals that focus on ecosystems, biodiversity and agriculture. This is initially due to the different ages or maturity of the ontology portals, while newer portals have already integrated the calculation of the FAIR score.

OntoPortal [Jo23] is an open source software for managing ontology repositories. The OntoPortal Alliance is a company and a group of research and infrastructure teams that support the development of ontology repositories in science and other fields. Examples of branches of this technology are **BioPortal**, the most important and historical source of the OntoPortal code, as well as the domain portals **AgroPortal**, **BiodivPortal**, **EarthPortal** and **EcoPortal**.

BioPortal [No09] focuses on the creation and provision of biomedical ontologies. The platform was developed and published by NCBO (National Center for Biomedical Ontology).

Tab. 1: Ontology Portals in ecosystems, biodiversity, agriculture and related fields. Portals are sorted in descending order w.r.t. the number of available ontologies. (Status: November 21, 2024)

Portal	URL	Number of ontologies	FAIR score
BioPortal	https://bioportal.bioontology.org/	1.159	✗
AberOWL	http://aber-owl.net/#/	1.149	✗
FAIRsharing	https://fairsharing.org	749	✗
Bioregistry	https://bioregistry.io	419	✗
OLS	https://www.ebi.ac.uk/ols/index	266	✗
Ontobee	https://ontobee.org/	265	✗
OBO Foundry	https://obofoundry.org/	261	✗
AgroPortal	https://agroportal.lirmm.fr/	234	✓
BiodivPortal	https://biodivportal.gfbio.org/	45	✓
EarthPortal	http://earthportal.eu/	42	✓
EcoPortal	https://ecoportal.lifewatch.eu/	35	✓

The aim of the portal is to give the community the opportunity to participate in the development and evaluation of ontologies.

AgroPortal [Jo18] stores ontologies and metadata in the field of agriculture and food. The scope of AgroPortal includes agricultural sciences, plants, nutrition and environment. It is part of the D2KAB (Data to Knowledge in Agronomy and Biodiversity) project. AgroPortal is intended to play a similar role in agronomy, agriculture and biodiversity as Bioportal does in biomedicine.

BiodivPortal [Ka24] specializes on ontologies in the field of biodiversity and environmental sciences. The platform is part of the German NFDI (National Research Data Infrastructure) program which aims to offer comprehensive data management within research institutions on a national level. It intends to promote standardization in order to support FAIR terminologies and FAIR research data. An important aspect of the development is the federated access to taxonomies, which often first have to be converted into a Semantic Web-compliant format. A process was developed to realize this transfer and subsequent integration into BiodivPortal.

EarthPortal [APV23] emphasizes ontologies in the area of earth systems, e.g. atmosphere, land surfaces and solid earth. The portal wants to find a way to combine and match different semantic artifacts (e.g. vocabularies, ontologies) effectively and FAIR. To achieve this, an ontology portal for storing and cataloging semantic artifacts is provided to simplify the mapping of concepts and connect research in this area.

The core topic of **EcoPortal** [Ke21] is ecology. It is an initiative of LifeWatch ERIC (European Research Infrastructure Consortium), the European digital science and technology infrastructure for biodiversity and ecosystem research. Ecoportal implements a FAIR Knowledge Organization System (FAIR KOS) to advance the homogenization of the growing data stream, e.g. by regulating the use of metadata, mappings and citations.

In addition to the ontology portals of the OntoPortal family, there are other portals such as **Aber-OWL**, **Bioregistry**, **FAIRsharing**, **Ontology Lookup Service**, **Ontobee** and **OBO Foundry**. With the exception of FAIRsharing, they focus on the fields of biology and biomedicine. Some of these platforms are also referred to as registries, frameworks or repositories and offer additional metadata to support research. OBO Foundry, for example, enables the thematic grouping of ontologies, which simplifies exploration. Bioregistry also contains a meta-registry, i.e. an overview of the occurrence of ontologies in other portals. FAIRsharing contains a collection of guidelines, data and metadata standards and databases.

4 Ontology Overview

For an overview of the ontology landscape in the fields of ecosystems, biodiversity and agriculture, we first collected ontologies from a comprehensive search using Google, Google Scholar and the Digital Bibliography & Library Project (DBLP). The queries were composed of the terms *Agriculture*, *Ecological*, *Ecosystem*, *Climate protection* and *Biodiversity* (as well as their German equivalents), each in combination with the terms *Ontology* and *Knowledge Graph*. We further used semantic links of the Linked Open Data Cloud to enrich the results. In total, the search results in 41 relevant ontologies. We provide a tabular overview in Table 2 to summarize the collected ontologies according to several categories: available formats, indicators, their availability on the relevant ontology portals from Table 1, number of classes/individuals/properties and their primary publication. An earlier version of the table has been created as part of a master's thesis [Ha24]. For clarity, we use the acronyms of the ontologies in the table (full names are listed in Table 3 in the Appendix). Furthermore, we show four indicators to compare the ontologies according to the year of last update, the number of visits, the development status and the FAIR score:

Year indicates the latest update of the ontology. If the ontology has a public version management, the year of the latest version is used. If the versions are only specified on the portals, the first upload of the latest version is used.

Visits indicates how many pageviews the ontology had on average per month in the last three years, starting January 1, 2022. The statistic is used for the calculation of the so called 'ontology acceptance' in the NCBO Ontology Recommender [Ma17]. AgroPortal, BioPortal, EcoPortal, BiodivPortal and EarthPortal keep these statistics and were used to collect this information. If more than one portal reports the number of visits, we sum the average values of monthly visits although the sets of page visitors might overlap.

Status indicates the development status of the ontology according to the meta information in AgroPortal and BioPortal: In production (++), in development as beta version (+) or alpha version (-) or retired (--). If different stages of development are indicated on the portals, the most advanced one is selected.

FAIR score reflects the extent to which a semantic resource takes the FAIR aspects (Findable, Accessible, Interoperable, Reusable) into account. The score is based on the O'FAIRe

(Ontology FAIRness Evaluator) method [AJ22, ABJ22]. For the calculation, an ontology is subjected to several automated tests for each aspect. The FAIR score increases for each test that is passed. A score of 100 would correspond to optimal compliance with the FAIR aspects. A detailed description of the individual aspects can be found on the portals. Not all ontology portals have currently implemented the FAIR Score Evaluation. In this overview, it is implemented by AgroPortal, BiodivPortal, EarthPortal and EcoPortal. If more than one portal saves the FAIR score of a given ontology, the value is averaged.

Tab. 2: Overview of the collected ontologies. The entries are sorted alphabetically w.r.t. the acronym of the ontology. A field is empty if no further information has been provided by the ontology portals. (Status: December 6, 2024)

	Formats				Indicators				Portals													
Name	SKOS	RDF/XML	OWL	OBO	Turtle	Year	Visits	Status	FAIR score (%)	AgroPortal	BioPortal	EcoPortal	Ontobee	OBO Foundry	AberOWL	OLS	Bioregistry	FAIRsharing	BiodivPortal	EarthPortal	**Classes**	**Individuals**	**Properties**	**Article**
AFO		x	x		x	2018	25.4	++	50	●								●			7	31.991	2	[LS14]
AGRO			x	x		2022	60.5	++	69	●	●		●	●	●	●	●	●			4.163	552	209	[Au17]
AGROVOC		x	x			2024	149.4	++		●		●					●	●			34	1.281.468	209	[RK12]
AHOL			x	x		2024	12.9	-	58	●	●				●			●	●		1.104	0	11	[Le23]
ANAEETHES		x	x			2020	58.8	-	59	●								●			2	3.247	0	[Ch14]
ATOL			x	x		2024	24.8	++	61	●	●				●		●	●	●		2.352	0	0	[Hu14]
BCO			x	x		2021	23.1	++	58	●	●	●	●	●	●	●	●	●	●		254	64	378	[Wa14]
BERO			x	x		2023	113.7	-			●				●						392.307	299	302	[Am24]
BIODIVTHES		x	x		x	2022	27.4	++	59	●	●	●			●				●		4	915	2	[VGG18]
CGO			x	x	x	2024	18	++	57	●											1.357	2.638	409	[Ba21]
CO*			x							●							●	●						[Ar16]
DEMETER-AIM			x	x	x	2023	51.9	++	58	●											182	137	298	[Ro22]
DSW			x	x		2022	3.7	++	55	●		●									19	0	28	[BW16]
DWC			x			2024												●			-	-	-	[Wi12]
ECOCORE			x	x	x	2022	5.6	-			●		●	●	●	●	●				5.586	17	295	[Bu17]
ECSO			x	x		2024	94.8	+	51		●	●					●	●			2.094	18	122	[Ch24]
ECTO			x	x	x	2023	8.5	++	60	●	●		●	●	●	●	●				11.865	60	217	[Ch23]
ENVO			x	x	x	2024	177.7	++	68	●	●		●	●	●	●	●	●	●		7.030	50	167	[Bu13]
ENVTHES		x	x		x	2024	49.3	++	58		●	●			●			●			6	5.841	0	[SPB13]
EOL			x	x		2024	6.5	++	60	●	●				●		●	●	●		690	1.380	0	[Hu14]
EUROVOC		x	x			2024	5.9		46			●						●			2	7489	0	[Un95]
FLOPO			x	x		2019	11.2	+	64	●	●		●	●	●	●	●	●	●		29.020	0	138	[Ho16]
FOODON			x	x		2024	104.2	-	67	●	●		●	●	●	●	●	●			39.435	435	194	[Do18]
GACS		x	x			2018	19.7	--	53	●								●			12	584.881	3	[Ba16]
GEMET		x	x			2021	22.8	++	48	●							●	●		●	7	5.739	0	[Pu98]
IOBC			x	x	x	2021	212.7	+			●				●		●				126.847	66.923	59	[Ku19]
MEAT-T		x	x			2022	23.4	-	55	●											2	1.505	0	[Ko22]
NALT		x	x		x	2024	26.6	++	59	●								●			8	83.644	6	[Yo08]
NCBITAXON			x	x	x	2024	465.2	++	60	●	●		●	●	●	●	●	●	●		707.357	707.351	22	[Wh07]
OBOE			x	x		2019	61.7	++	58	●	●	●			●			●	●		292	0	31	[Ma07]
ONTOBIOTOPE			x		x	2021	43.2	++	58	●								●			4.219	0	1	[Né18]
PCO			x	x	x	2021	15.5	-	59	●	●	●	●	●	●	●	●	●			221	18	110	[Wa14]
PEAO			x	x		2019	4.7	++	56	●	●				●			●			2.298	1	74	[Me14]
PECO (ex-EO)			x	x	x	2023	33.6	++	71	●	●		●	●	●	●	●	●			3.136	0	119	[Ja02]
PO			x	x	x	2024	99.4	++	74	●	●		●	●	●	●	●	●	●		2.026	9	125	[Co02]

Continued on next page

Continued from previous page

	Formats				Indicators				Portals													
Name	SKOS	RDF/XML	OWL	OBO	Turtle	Year	Visits	Status	FAIR score (%)	AgroPortal	BioPortal	EcoPortal	Ontobee	OBO Foundry	AberOWL	OLS	Bioregistry	FAIRsharing	BiodivPortal	EarthPortal	**Classes**	**Individuals**	**Properties**	**Article**
PPO		x	x	x		2024	3.6	++	68	●	●		●	●	●	●	●	●			543	240	147	[St18]
PSO/PLANTSO		x	x	x		2023	5.6	++	68	●	●		●	●	●	●	●				3.933	9	138	[Co17]
SDGIO		x	x	x		2018	4.1	++	56	●			●				●	●			907	470	152	[SJ16]
SWEET		x	x		x	2022	122.8	++	50		●				●		●	●	●	●	10.239	2.148	359	[RP05]
TAXREF-LD	x	x	x		x	2024	12.4	++	65	●											287.229	1.019.653	6	[Mi17]
TO/PTO		x	x	x		2023	132.5	++	73	●	●		●	●	●	●	●	●	●		5.290	9	173	[Co02]
Ontology count										33	25	9	15	14	24	14	23	30	12	2				

* The Crop Ontology consists of several sub-ontologies, the individual ontologies have their own releases, formats, indicators, classes, individuals and properties.

Table 2 lists relevant ontologies for an interdisciplinary project in the context of ecosystems, biodiversity and agriculture such as “AgriRestore”. Most of the ontologies examined (36/41) appear in more than one ontology portal. 5 ontologies are available on only one portal. Furthermore, 16 ontologies are available on 2 to 4 portals. The ontology with the highest availability (BCO) is available on 10 of the 11 portals analyzed. This is followed by a larger number of ontologies that are available on 7 to 9 of the portals (12). Regarding the availability of ontologies on the portals, the majority of the examined ontologies (33) can be found on AgroPortal. Although AgroPortal does not contain the largest overall number of ontologies (see Table 1), it concentrates on ontologies from the agricultural sector and plant life. In contrast, BioPortal contains the highest number of ontologies, but less of the ontologies of interest (25) possibly due to its specialization on biomedical ontologies. 19 ontologies overlap in the two portals. BiodivPortal and EarthPortal contain the smallest number of ontologies, probably due to the novelty of these portals. The table further shows that all ontologies are available in the RDF/XML format, a large number is available for download in the OWL format, while the formats SKOS, OBO and Turtle only appear occasionally. The availability of all ontologies in RDF/XML format shows that it is an accepted standard for the provision of ontologies. The majority of the ontologies are still actively maintained and have been updated within the last three years. According to their own statement, most ontologies are in productive status. As this statement is self-defined, it does not always comply with the latest update. The number of pageviews per month within the last three years ranges from less than 5 to almost 470 on average indicating that some ontologies are already heavily integrated into processes or are supported by large organizations. Finally, some portals support the calculation of the FAIR score; AgroPortal in particular provides it for most of the considered ontologies. Even if all ontologies should support the FAIR principles, the maximum is reached at 73% and drops to a minimum of 46%, so an optimum has not yet been reached.

5 Open Challenges

Ontology portals usually collect ontologies for specific purposes and provide a good starting point for finding suitable ontologies for a research project in the context. An in-depth analysis of the existing ontology landscape is required to achieve high RDM quality, e.g. to reuse existing concepts and relationships when developing new ontologies. Still, this is a challenging and time-consuming task given that relevant ontologies for interdisciplinary projects are spread across different portals and a single portal may not fit all requirements. Multiple portals covering similar ontologies offer users alternatives to choose a platform that better suits their specific needs. However, the **redundant provision on several portals** leads to distribution and duplicated efforts which may result in incompatibility and hinder the findability and reusability of ontologies. Effective **coordination and cooperation** of ontology portal providers is necessary to mitigate these issues and maximize the usability and accessibility of ontologies that fit the purposes of several ontology portals. Despite the aim of ontology portals to collect all ontologies of a domain, it is difficult to maintain such a collection, especially in view of the **growing number of ontologies**. In addition, there are the **various formats** that strive for standardization but may complicate integration. Ontology portals already collect various metrics and statistics that could be further extended to support assessing the usefulness of an ontology. This could include the actual use of ontologies for the annotation of content in databases or an indicator that shows the extent to which ontologies provide a rich model allowing inference beyond sub/super concepts. Both pieces of information would emphasize the relevance of an ontology.

Domain experts and ontology developers aim to create ontologies that accurately represent real-world concepts and relationships. Using these ontologies supports good RDM and facilitates data integration. Ontologies and their links to data sets and other ontologies are continuously extended and updated [GPR16], requiring the detection and monitoring of **ontology changes** [HGR13] as well as the migration of outdated artifacts [Gr13]. In addition to quality criteria such as the accuracy and consistency of an ontology, its findability, accessibility, interoperability and reusability are of central importance. They and can be evaluated using the **FAIR score**, which should be supported by all ontology portals and encourage ontology maintainers to make improvements.

As the number of ontologies increases, so does the need for them and for knowledge graphs. Manual creation is being replaced more and more by **(semi-)automatic ontology learning** based on scientific documents such as publications. This process has been further boosted by the integration of Large Language Models (LLMs) into ontology learning, so that new knowledge graphs and ontologies can be created automatically on the basis of a corpus suitable for a project. The advantages of reliable LLMs in ontology learning could include speed and the automatic identification of terms, types and relations [BGDA23]. Current LLMs still lack the logical reasoning and expertise, especially in complex domains such as life science, to construct reliable ontologies [FSA24]. The process requires meaningful contextual information and the involvement of domain experts for **verification of automatic recommendations**.

Overall, ontology search and development face the challenges of efficiently managing a

growing number of evolving ontologies and diverse formats for different but overlapping scientific communities, while ensuring coordination and high quality through cooperation and reliable automated processes. This goes along with finding better and more efficient ways for validation and compliance with FAIR principles in existing and emerging ontologies as well as knowledge graphs that reuse and integrate the ontologies.

6 Conclusions and Future Directions

After a brief introduction to ontologies and their formats, we summarized ontology portals and ontologies relevant to interdisciplinary ecosystem, biodiversity and agriculture research and discussed open challenges for ontology search and development. Research at the interface of these domains is essential for the long-term maintenance of ecosystem services for human well-being, such as food production [Mea05]. In this context, knowledge graphs and ontologies help to achieve a holistic understanding of the complex processes and relationships. To get an overview of the ontology landscape, we summarized relevant ontologies from the fields of ecosystems, biodiversity and agriculture based on categories such as available formats, their FAIR score and recency. Currently, many ontologies are distributed and partly redundantly provided by different ontology portals hindering the actual intention to simplify ontology search and querying on one central platform.
Our search further revealed that there is currently no ontology focusing on ecosystem restoration measures, particularly in terms of their impact on agriculture and ecosystem services. In the future, we plan to develop a suitable ontology based on the reuse of existing ontologies where possible. Calculating the semantic overlap of ontologies can support the selection process and help to reduce redundancies. The ontologies will be used to develop novel knowledge graphs for comprehensive analyses in interdisciplinary projects such as "AgriRestore".

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Project-ID 528485254 - FIP 16 and by the Europäischer Fonds für regionale Entwicklung (EFRE, European Regional Development Fund) Project number ZS/2023/12/182018.

Bibliography

[ABJ22] Amdouni, Emna; Bouazzouni, Syphax; Jonquet, Clement: O'FAIRe: Ontology FAIRness Evaluator in the AgroPortal Semantic Resource Repository. In: The Semantic Web: ESWC 2022 Satellite Events. Springer International Publishing, pp. 89–94, 2022.

[AJ22] Amdouni, Emna; Jonquet, Clement: FAIR or FAIRer? An Integrated Quantitative FAIRness Assessment Grid for Semantic Resources and Ontologies. In: Metadata and Semantic Research. Springer International Publishing, pp. 67–80, 2022.

[Am24] Amusat, Oluwamayowa O; Hegde, Harshad; Mungall, Christopher J; Giannakou, Anna; Byers, Neil P; Gunter, Dan; Fagnan, Kjiersten; Ramakrishnan, Lavanya: Automated annotation of scientific texts for ML-based keyphrase extraction and validation. Database, 2024, 2024.

[APV23] Alviset, Guillaume; Pierkot, Christelle; Vernet, Marine: The EarthPortal, towards an ontology repository for the Earth System semantic artefacts. In: Joint Proceedings of the Onto4FAIR 2023 Workshops. p. 17, 2023.

[Ar16] Arnaud, Elizabeth; Valette, L; Pietragalla, J; Laporte, Marie-Angélique; Aubert, C; Devare, Medha; McLaren, G; Ribaut, JM: The Crop Ontology: a source of standard traits and variables for breeding and agronomy. Bioversity Conference Papers, 2016.

[Au17] Aubert, C.; Buttigieg, P. L.; Laporte, M. A.; Devare, M.; Arnaud, E.: CGIAR Agronomy Ontology. CGIAR, 2017.

[Ba16] Baker, Thomas; Caracciolo, Caterina; Doroszenko, Anton; Suominen, Osma: GACS Core: Creation of a Global Agricultural Concept Scheme. In: Metadata and Semantics Research. Springer International Publishing, pp. 311–316, 2016.

[Ba21] Bakker, Roos; van Drie, Romy; Bouter, Cornelis; van Leeuwen, Sander; van Rooijen, Lorijn; Top, Jan: The Common Greenhouse Ontology: An Ontology Describing Components, Properties, and Measurements inside the Greenhouse. In: The 13th EFITA International Conference. EFITA 2021. MDPI, p. 27, December 2021.

[BGDA23] Babaei Giglou, Hamed; D'Souza, Jennifer; Auer, Sören: LLMs4OL: Large Language Models for Ontology Learning. In: The Semantic Web – ISWC 2023. Springer Nature Switzerland, pp. 408–427, 2023.

[BLHL23] Berners-Lee, Tim; Hendler, James; Lassila, Ora: The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities. In: Linking the World's Information. ACM, pp. 91–103, July 2023.

[BM04] Beckett, Dave; McBride, Brian: RDF/XML syntax specification (revised). W3C recommendation, 10(2.3), 2004.

[Bu13] Buttigieg, Pier; Morrison, Norman; Smith, Barry; Mungall, Christopher J; Lewis, Suzanna E: The environment ontology: contextualising biological and biomedical entities. Journal of Biomedical Semantics, 4(1):43, 2013.

[Bu17] Buttigieg, Pier Luigi; Mungall, Chris; Blumberg, Kai; Marie-Angélique Laporte; , Renzo; Uberon: EnvironmentOntology/envo: Planetary ecology, 2017.

[BVH18] Bommarco, Riccardo; Vico, Giulia; Hallin, Sara: Exploiting ecosystem services in agriculture for increased food security. Global food security, 17:57–63, 2018.

[BW16] Baskauf, Steven J.; Webb, Campbell O.: Darwin-SW: Darwin Core-based terms for expressing biodiversity data as RDF. Semantic Web, 7(6):629–643, October 2016.

[Ca20] Carriero, Valentina Anita; Daquino, Marilena; Gangemi, Aldo; Nuzzolese, Andrea Giovanni; Peroni, Silvio; Presutti, Valentina; Tomasi, Francesca: The Landscape of Ontology Reuse Approaches. In: Applications and Practices in Ontology Design, Extraction, and Reasoning. IOS Press, pp. 21–38, November 2020.

[Ch14] Chanzy, Andre; Pichot, Christian; Callou, Cécile; Lafolie, Francois; Loreau, Michel; Raynal, Helene; Greiveldinger, Lucile: Integrating modelling and experimental platforms in research infrastructure: design and approach in AnaEE-France. In: ExpeER international conference: Experimentation in Ecosystem Research in a changing world: Challenges and opportunities. p. 1, 2014.

[Ch23] Chan, Lauren E.; Thessen, Anne E.; Duncan, William D.; Matentzoglu, Nicolas; Schmitt, Charles; Grondin, Cynthia J.; Vasilevsky, Nicole; McMurry, Julie A.; Robinson, Peter N.; Mungall, Christopher J.; Haendel, Melissa A.: The Environmental Conditions, Treatments, and Exposures Ontology (ECTO): connecting toxicology and exposure to human health and beyond. Journal of Biomedical Semantics, 14(1), February 2023.

[Ch24] Chong, Steven S.; Schildhauer, Mark; O'Brien, Margaret; Mecum, Bryce; Jones, Matthew B.: Enhancing the FAIRness of Arctic Research Data Through Semantic Annotation. Data Science Journal, 23, 2024.

[Co02] Consortium, The Plant Ontology™: The Plant Ontology™ Consortium and Plant Ontologies. Comparative and Functional Genomics, 3(2):137–142, April 2002.

[Co12] Consortium, World Wide Web et al.: OWL 2 web ontology language document overview. W3C Recommendation, 2012.

[Co14] Consortium, World Wide Web et al.: RDF 1.1 Turtle: terse RDF triple language. W3C Recommendation, 2014.

[Co17] Cooper, Laurel; Meier, Austin; Laporte, Marie-Angélique; Elser, Justin L; Mungall, Chris; Sinn, Brandon T; Cavaliere, Dario; Carbon, Seth; Dunn, Nathan A; Smith, Barry; Qu, Botong; Preece, Justin; Zhang, Eugene; Todorovic, Sinisa; Gkoutos, Georgios; Doonan, John H; Stevenson, Dennis W; Arnaud, Elizabeth; Jaiswal, Pankaj: The Planteome database: an integrated resource for reference ontologies, plant genomics and phenomics. Nucleic Acids Research, 46(D1):D1168–D1180, November 2017.

[Do18] Dooley, Damion M.; Griffiths, Emma J.; Gosal, Gurinder S.; Buttigieg, Pier L.; Hoehndorf, Robert; Lange, Matthew C.; Schriml, Lynn M.; Brinkman, Fiona S. L.; Hsiao, William W. L.: FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Science of Food, 2(1), December 2018.

[Fi24] Fischer, Christina; Jörg, Geistlinger; Groß, Anika; Kirmer, Anita; Knauer, Uwe; Meyer, Markus; Pause, Marion; Rozhon, Wilfried; Tischew, Sabine: AgriRestore: Ecosystem and landscape restoration across spatial and temporal scales to enhance biodiversity and climate resilience in agricultural landscapes, 04 2024.

[FSA24] Fathallah, Nadeen; Staab, Steffen; Algergawy, Alsayed: LLMs4Life: Large Language Models for Ontology Learning in Life Sciences, 2024.

[GPR16] Groß, Anika; Pruski, Cédric; Rahm, Erhard: Evolution of biomedical ontologies and mappings: overview of recent approaches. Computational and structural biotechnology journal, 14:333–340, 2016.

[Gr93] Gruber, Thomas R.: A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, June 1993.

[Gr13] Groß, Anika; Dos Reis, Julio Cesar; Hartung, Michael; Pruski, Cédric; Rahm, Erhard: Semi-automatic Adaptation of Mappings between Life Science Ontologies. In (Baker, Christopher J. O.; Butler, Greg; Jurisica, Igor, eds): Data Integration in the Life Sciences. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 90–104, 2013.

[Ha24] Hauke, Thorsten: Entwicklung eines integrierten Wissensgraphen aus Ontologien für Biodiversität und Landwirtschaft. Masterthesis, 2024.

[HGR13] Hartung, Michael; Groß, Anika; Rahm, Erhard: COnto–Diff: generation of complex evolution mappings for life science ontologies. Journal of Biomedical Informatics, 46(1):15–32, 2013.

[Hi08] Hitzler, Pascal; Krötzsch, Markus; Rudolph, Sebastian; Sure-Vetter, York, eds. Semantic Web. SpringerLink. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Hi21] Hitzler, Pascal: A review of the semantic web field. Communications of the ACM, 64(2):76–83, January 2021.

[Ho16] Hoehndorf, Robert; Alshahrani, Mona; Gkoutos, Georgios V.; Gosline, George; Groom, Quentin; Hamann, Thomas; Kattge, Jens; de Oliveira, Sylvia Mota; Schmidt, Marco; Sierra, Soraya; Smets, Erik; Vos, Rutger A.; Weiland, Claus: The flora phenotype ontology (FLOPO): tool for integrating morphological traits and phenotypes of vascular plants. Journal of Biomedical Semantics, 7(1), November 2016.

[Ho24] Hofer, Marvin; Obraczka, Daniel; Saeedi, Alieh; Köpcke, Hanna; Rahm, Erhard: Construction of Knowledge Graphs: Current State and Challenges. Information, 15(8):509, August 2024.

[Hu14] Hue, I.; Bugeon, J.; Dameron, O.; Fatet, A.; Hurtaud, C.; Joret, L.; Meunier-Salaün, M.C.; Nédellec, C.; Reichstadt, M.; Vernet, J.; Le Bail, P.Y.: ATOL and EOL ontologies, steps towards embryonic phenotypes shared worldwide? Animal Reproduction Science, 149(1–2):99, September 2014.

[Ja02] Jaiswal, Pankaj; Ware, Doreen; Ni, Junjian; Chang, Kuan; Zhao, Wei; Schmidt, Steven; Pan, Xiaokang; Clark, Kenneth; Teytelman, Leonid; Cartinhour, Samuel; Stein, Lincoln; McCouch, Susan: Gramene: development and integration of trait and gene ontologies for rice. Comparative and Functional Genomics, 3(2):132–136, April 2002.

[Ji22] Ji, Shaoxiong; Pan, Shirui; Cambria, Erik; Marttinen, Pekka; Yu, Philip S.: A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2):494–514, February 2022.

[Jo18] Jonquet, Clément; Toulet, Anne; Arnaud, Elizabeth; Aubin, Sophie; Dzalé Yeumo, Esther; Emonet, Vincent; Graybeal, John; Laporte, Marie-Angélique; Musen, Mark A.; Pesce, Valeria; Larmande, Pierre: AgroPortal: A vocabulary and ontology repository for agronomy. Computers and Electronics in Agriculture, 144:126–143, January 2018.

[Jo23] Jonquet, Clement; Graybeal, John; Bouazzouni, Syphax; Dorf, Michael; Fiore, Nicola; Kechagioglou, Xeni; Redmond, Timothy; Rosati, Ilaria; Skrenchuk, Alex; Vendetti, Jennifer L.; Musen, Mark: Ontology Repositories and Semantic Artefact Catalogues with the OntoPortal Technology. In: The Semantic Web – ISWC 2023. Springer Nature Switzerland, pp. 38–58, 2023.

[Ka20] Karam, Naouel; Khiat, Abderrahmane; Algergawy, Alsayed; Sattler, Melanie; Weiland, Claus; Schmidt, Marco: Matching biodiversity and ecology ontologies: challenges and evaluation results. The Knowledge Engineering Review, 35, 2020.

[Ka24] Karam, Naouel; Fillies, Jan; Jonquet, Clement; Bouazzouni, Syphax; Löffler, Felicitas; Zander, Franziska; König-Ries, Birgitta; Güntsch, Anton; Diepenbroek, Michael; Paschke, Adrian: BiodivPortal: Enabling Semantic Services for Biodiversity within the German National Research Data Infrastructure. Datenbank-Spektrum, 24(2):129–137, June 2024.

[Ke21] Kechagioglou, Xeni; Vaira, Lucia; Tomassino, Pierfrancesco; Fiore, Nicola; Basset, Alberto; Rosati, Ilaria: EcoPortal: An Environment for FAIR Semantic Resources in the Ecological Domain. In: JOWO. 2021.

[Ko22] Kombolo Ngah, Moise; Yon, Jérémy; Landrieu, François; Richon, Brigitte; Aubin, Sophie; Hocquette, Jean-François: A new semantic resource responding to the principles of Open Science: The meat thesaurus as an IT tool for dialogue between sector actors. Meat Science, 192:108849, October 2022.

[Ku19] Kushida, Tatsuya; Kozaki, Kouji; Kawamura, Takahiro; Tateisi, Yuka; Yamamoto, Yasunori; Takagi, Toshihisa: Interconnection of Biological Knowledge Using NikkajiRDF and Interlinking Ontology for Biological Concepts. New Generation Computing, 37(4):525–549, September 2019.

[Le23] Leblanc-Maridor, M.; Ferret, C.; Meunier-Salaün, M.C.; Gunia, M.; Reichstadt, M.; Le Floc'h, N.: 107. The AHOL (Animal Health Ontology for Livestock) ontology for integrating data on the main diseases of farmed animals: An example in the pig sector. Animal - science proceedings, 14(5):699–700, August 2023.

[LG12] Lanthaler, Markus; Gütl, Christian: On using JSON-LD to create evolvable RESTful services. In: Proceedings of the Third International Workshop on RESTful Design. WS-REST '12. ACM, April 2012.

[LM01] Lassila, Ora; McGuinness, Deborah: The role of frame-based representation on the semantic web. Linköping University Electronic Press, 2001.

[LS14] Lipsanen, Päivi; Siirala, Esko: AFO: asiasanastosta ontologiaksi. Informaatiotutkimus, 33(3), 2014.

[Ma07] Madin, Joshua; Bowers, Shawn; Schildhauer, Mark; Krivov, Sergeui; Pennington, Deana; Villa, Ferdinando: An ontology for describing and synthesizing ecological observation data. Ecological Informatics, 2(3):279–296, October 2007.

[Ma15] Mantyka-Pringle, Chrystal S; Visconti, Piero; Di Marco, Moreno; Martin, Tara G; Rondinini, Carlo; Rhodes, Jonathan R: Climate change modifies risk of global biodiversity loss due to land-cover change. Biological Conservation, 187:103–111, 2015.

[Ma17] Martínez-Romero, Marcos; Jonquet, Clement; O'Connor, Martin J.; Graybeal, John; Pazos, Alejandro; Musen, Mark A.: NCBO Ontology Recommender 2.0: an enhanced approach for biomedical ontology recommendation. Journal of Biomedical Semantics, 8(1), June 2017.

[MB09] Miles, Alistair; Bechhofer, Sean: SKOS simple knowledge organization system reference. W3C recommendation, 2009.

[Me14] Mendes, Nuno D; Monteiro, Pedro T; Vaz, Cátia; Chaves, Inês: Towards a plant experimental assay ontology. DILS 2014, p. 41, 2014.

[Mea05] Millennium ecosystem assessment, MEA: Ecosystems and human well-being, volume 5. Island press Washington, DC, 2005.

[Mi17] Michel, Franck; Gargominy, Olivier; Tercerie, Sandrine; Zucker, Catherine Faron: A model to represent nomenclatural and taxonomic information as linked data. Application to the French Taxonomic Register, TAXREF. In: ISWC 2017 Workshop on Semantics for Biodiversity (S4Biodiv 2017). pp. 1–12, 2017.

[Né18] Nédellec, Claire; Bossy, Robert; Chaix, Estelle; Deléger, Louise: Text-mining and ontologies: new approaches to knowledge discovery of microbial diversity. arXiv preprint arXiv:1805.04107, pp. 353–358, 2018.

[No09] Noy, N. F.; Shah, N. H.; Whetzel, P. L.; Dai, B.; Dorf, M.; Griffith, N.; Jonquet, C.; Rubin, D. L.; Storey, M.-A.; Chute, C. G.; Musen, M. A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research, 37(Web Server):W170–W173, May 2009.

[Po10] Power, Alison G: Ecosystem services and agriculture: tradeoffs and synergies. Philosophical transactions of the royal society B: biological sciences, 365(1554):2959–2971, 2010.

[Pu98] Purat, Jacek: The world of multilingual environmental thesauri. Search Support for Unfamiliar Metadata Vocabularies, 1998.

[RK12] Rajbhandari, Sachit; Keizer, Johannes: The AGROVOC Concept Scheme – A Walkthrough. Journal of Integrative Agriculture, 11(5):694–699, May 2012.

[Ro22] Routis, George; Paraskevopoulos, Marios; Vetsikas, Ioannis A.; Roussaki, Ioanna; Stavrakoudis, Dimitris; Katsantonis, Dimitrios: Data-Driven and Interoperable Smart Agriculture: An IoT-based Use-Case for Arable Crops. In: 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). IEEE, Piscataway, NJ, pp. 1–8, August 2022. Literaturangaben.

[RP05] Raskin, Robert G.; Pan, Michael J.: Knowledge representation in the semantic web for Earth and environmental terminology (SWEET). Computers & Geosciences, 31(9):1119–1125, November 2005.

[Sa21] Sakr, Sherif; Bonifati, Angela; Voigt, Hannes; Iosup, Alexandru; Ammar, Khaled; Angles, Renzo; Aref, Walid; Arenas, Marcelo; Besta, Maciej; Boncz, Peter A.; Daudjee, Khuzaima; Valle, Emanuele Della; Dumbrava, Stefania; Hartig, Olaf; Haslhofer, Bernhard; Hegeman, Tim; Hidders, Jan; Hose, Katja; Iamnitchi, Adriana; Kalavri, Vasiliki; Kapp, Hugo; Martens, Wim; Özsu, M. Tamer; Peukert, Eric; Plantikow, Stefan; Ragab, Mohamed; Ripeanu, Matei R.; Salihoglu, Semih; Schulz, Christian; Selmer, Petra; Sequeda, Juan F.; Shinavier, Joshua; Szárnyas, Gábor; Tommasini, Riccardo; Tumeo, Antonino; Uta, Alexandru; Varbanescu, Ana Lucia; Wu, Hsiang-Yun; Yakovets, Nikolay; Yan, Da; Yoneki, Eiko: The future is big graphs: a community view on graph processing systems. Communications of the ACM, 64(9):62–71, August 2021.

[SBF98] Studer, Rudi; Benjamins, V.Richard; Fensel, Dieter: Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2):161–197, March 1998.

[SJ16] Smith, Barry; Jensen, Mark: The UNEP Ontologies and the OBO Foundry. In: ICBO/BioCreative. 2016.

[SPB13] Schentz, Herbert; Peterseil, Johannes; Bertrand, Nic: EnvThes – interlinked thesaurus for long term ecological research, monitoring, and experiments. In: Proceedings of the 27th Conference on Environmental Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management. Shaker Verlag, Aachen, 2013.

[St18] Stucky, Brian J.; Guralnick, Rob; Deck, John; Denny, Ellen G.; Bolmgren, Kjell; Walls, Ramona: The Plant Phenology Ontology: A New Informatics Resource for Large-Scale Integration of Plant Phenology Data. Frontiers in Plant Science, 9, May 2018.

[Ti11] Tirmizi, Syed; Aitken, Stuart; Moreira, Dilvan A; Mungall, Chris; Sequeda, Juan; Shah, Nigam H; Miranker, Daniel P: Mapping between the OBO and OWL ontology languages. Journal of Biomedical Semantics, 2(Suppl 1):S3, 2011.

[Ti16] Titeux, Nicolas; Henle, Klaus; Mihoub, Jean-Baptiste; Regos, Adrián; Geijzendorffer, Ilse R; Cramer, Wolfgang; Verburg, Peter H; Brotons, Lluís: Biodiversity scenarios neglect future land-use changes. Global change biology, 22(7):2505–2515, 2016.

[Un95] Union, European: Thesaurus Eurovoc. Publications Office, Luxembourg, edition 3 edition, 1995.

[VGG18] Vachez, Dominique; Gomez, Isabelle; Garnier, Eric: Semantics of Biodiversity: from Thesaurus to Linked Open Data (LOD). In: 47th LIBER Annual Conference: Research Libraries as an Open Science Hub. 2018.

[Wa14] Walls, Ramona L.; Deck, John; Guralnick, Robert; Baskauf, Steve; Beaman, Reed; Blum, Stanley; Bowers, Shawn; Buttigieg, Pier Luigi; Davies, Neil; Endresen, Dag; Gandolfo, Maria Alejandra; Hanner, Robert; Janning, Alyssa; Krishtalka, Leonard; Matsunaga, Andréa; Midford, Peter; Morrison, Norman; Tuama, Éamonn Ó.; Schildhauer, Mark; Smith, Barry; Stucky, Brian J.; Thomer, Andrea; Wieczorek, John; Whitacre, Jamie; Wooley, John: Semantics in Support of Biodiversity Knowledge Discovery: An Introduction to the Biological Collections Ontology and Related Ontologies. PLoS ONE, 9(3):e89606, March 2014.

[Wh07] Wheeler, D. L.; Barrett, T.; Benson, D. A.; Bryant, S. H.; Canese, K.; Chetvernin, V.; Church, D. M.; DiCuccio, M.; Edgar, R.; Federhen, S.; Feolo, M.; Geer, L. Y.; Helmberg, W.; Kapustin, Y.; Khovayko, O.; Landsman, D.; Lipman, D. J.; Madden, T. L.; Maglott, D. R.; Miller, V.; Ostell, J.; Pruitt, K. D.; Schuler, G. D.; Shumway, M.; Sequeira, E.; Sherry, S. T.; Sirotkin, K.; Souvorov, A.; Starchenko, G.; Tatusov, R. L.; Tatusova, T. A.; Wagner, L.; Yaschenko, E.: Database resources of the National Center for Biotechnology Information. Nucleic Acids Research, 36(Database):D13–D21, December 2007.

[Wi12] Wieczorek, John; Bloom, David; Guralnick, Robert; Blum, Stan; Döring, Markus; Giovanni, Renato; Robertson, Tim; Vieglais, David: Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. PLoS ONE, 7(1):e29715, January 2012.

[Wi16] Wilkinson, Mark D.; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Bonino; Bourne, Philip E.; Bouwman, Jildau; Brookes, Anthony J.; Clark, Tim; Crosas, Mercè; Dillo, Ingrid; Dumon, Olivier; Edmunds, Scott; Evelo, Chris T.; Finkers, Richard; Gonzalez-Beltran, Alejandra; Gray, Alasdair J.G.; Groth, Paul; Goble, Carole; Grethe, Jeffrey S.; Heringa, Jaap; 't Hoen, Peter A.C; Hooft, Rob; Kuhn, Tobias; Kok,

Ruben; Kok, Joost; Lusher, Scott J.; Martone, Maryann E.; Mons, Albert; Packer, Abel L.; Persson, Bengt; Rocca-Serra, Philippe; Roos, Marco; van Schaik, Rene; Sansone, Susanna-Assunta; Schultes, Erik; Sengstag, Thierry; Slater, Ted; Strawn, George; Swertz, Morris A.; Thompson, Mark; van der Lei, Johan; van Mulligen, Erik; Velterop, Jan; Waagmeester, Andra; Wittenburg, Peter; Wolstencroft, Katherine; Zhao, Jun; Mons, Barend: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3(1), March 2016.

[Yo08] Young, Peter R: The National Agricultural Library and E-Science. Journal of Information Processing and Management, 51(9):684–695, 2008.

Appendix

Tab. 3: Assignment of the names of ontologies to acronyms

Acronym	Name
AFO	Agriculture and Forestry Ontology
AGRO	AGRonomy Ontology
AGROVOC	Agronomy Vocabulary
AHOL	Animal Health Ontology for Livestock
AnaEEThes	Analysis and Experimentation on Ecosystems Thesaurus
ATOL	Animal Trait Ontology for Livestock
BCO	Biological Collections Ontology
BERO	Biological and Environmental Research Ontology
BIODIVTHES	Biodiversity Thesaurus
CGO	Common Greenhouse Ontology
CO	Crop Ontology
DEMETER-AIM	DEMETER Agriculture Information Model
DSW	Darwin Semantic Web
DWC	Darwin Core
ECOCORE	Ecological Core Entities
ECSO	Ecosystem Ontology
ECTO	Environmental conditions, treatments and exposures ontology
ENVO	Environment Ontology
ENVTHES	Thesaurus for long-term ecological research, monitoring, experiments
EOL	Environment Ontology for Livestock
EUROVOC	European Vocabularies Core Concepts
FLOPO	Flora Phenotype Ontology
FOODON	Food Ontology
GACS	Global Agricultural Concept Scheme
GEMET	GEneral Multilingual Environmental Thesaurus
IOBC	Interlinking Ontology for Biological Concepts
MEAT-T	Meat Thesaurus
NALT	National Agricultural Library Thesaurus
NCBITAXON	National Center for Biotechnology Information Taxonomy
OBOE	Extensible Observation Ontology
ONTOBIOTOPE	Biotope Ontology
PCO	Population and Community Ontology
PEAO	Plant Experimental Assay Ontology
PECO (ex-EO)	Plant Experimental Conditions Ontology
PO	Plant Ontology
PPO	Plant Phenology Ontology
PSO/PLANTSO	Plant Stress Ontology
SDGIO	Sustainable Development Goals Interface Ontology
SWEET	Semantic Web for Earth and Environment Technology Ontology
TAXREF-LD	Taxonomic reference for flora and fauna (of mainland France and overseas territories)
TO / PTO	Plant Trait Ontology

Semantic technologies for interdisciplinary research: A case study on improving data synthesis and integration in the biodiversity domain

Tarek Al Mustafa [1,3], Franziska Zander [1,2], David Schellenberger Costa [3,4], Erik Kleinsteuber [1,3], Christian Wirth [3,4], and Birgitta König-Ries [1,3]

Abstract: In biodiversity research, synthesizing data from different sources is frequently needed as a prerequisite to answering important questions. Performing this synthesis and integrating one's own research data remains a tedious process requiring significant human effort. Often, the results of these efforts are not easily reusable for other questions. Knowledge graphs have been proposed in the literature as an approach to alleviate this problem, in part through their inherent adherence to the FAIR data principles, but have gained little traction in biodiversity research practice so far due to significant challenges in knowledge graph construction and usage by non-domain experts. Our contribution showcases an approach and tools needed for knowledge graph creation, management, and usage implemented in the context of PlantHub and the iKNOW project. We present a knowledge graph combining plant trait sources within the PlantHub project (planthub.idiv.de) including preprocessed data from TRY, a plant trait database, with citizen science occurrence data from naturgucker.de and add taxonomic and additional information from multiple sources (e.g. Wikidata, GBIF, OpenElevation,...). We present the workflow needed to create such a graph and show different options for its management using features from Ontotext Refine for data cleaning, API & URL fetching, RDF mapping and export, and Ontotext GraphDB for hosting, querying, and visualization. To simplify usage of this graph, we showcase a query builder interface that allows users to construct SPARQL queries without needing any prior domain knowledge. We motivate our work and its application in the biodiversity research domain and contribute to bridging the gap towards using semantic technologies in this field of research.

Keywords: Knowledge Graphs, Ontologies, Data Synthesis, Data Integration, FAIR Data Principles, Biodiversity Research, Interdisciplinary Research

[1] Friedrich Schiller University Jena, Heinz Nixdorf Chair for Distibuted Information Systems, Leutragraben 1, 07743 Jena, Germany, tarek.almustafa@uni-jena.de, https://orcid.org/0000-0001-7793-4483; franziska.zander@uni-jena.de, https://orcid.org/0000-0001-6892-7046; erik.kleinsteuber@uni-jena.de, https://orcid.org/0000-0001-8388-4929; birgitta.koenig-ries@uni-jena.de, https://orcid.org/0000-0002-2382-9722

[2] Senckenberg Society for Nature Research, Senckenberg Institute for Plant Form and Function (SIP), Jena, Germany, franziska.zander@uni-jena.de, https://orcid.org/0000-0001-6892-7046

[3] German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig , Puschstr 4, 04103 Leipzig, Germany, tarek.almustafa@uni-jena.de, https://orcid.org/0000-0001-7793-4483; david.schellenberger.costa@uni-leipzig.de, https://orcid.org/0000-0003-1747-1506; erik.kleinsteuber@uni-jena.de, https://orcid.org/0000-0001-8388-4929; cwirth@uni-leipzig.de, https://orcid.org/0000-0003-2604-8056; birgitta.koenig-ries@uni-jena.de, https://orcid.org/0000-0002-2382-9722

[4] University of Leipzig, Department of Special Botany and Functional Biodiversity, Faculty of Life Sciences, Johannisallee 21-23, 04103 Leipzig, Germany, david.schellenberger.costa@uni-leipzig.de, https://orcid.org/0000-0003-1747-1506; cwirth@uni-leipzig.de, https://orcid.org/0000-0003-2604-8056

doi:10.18420/BTW2025-120

1 Introduction

There is massive ongoing biodiversity loss caused mostly by human impact on factors such as climate change [Sh22, Ha22], pollution of air, soil, and water, hunting and overexploitation, natural disasters, habitat loss, and others [SSS21]. To tackle this crisis, biodiversity research aims to quantify these effects, understand the underlying processes, and develop strategies to mitigate biodiversity loss and ecosystem degradation. This is not possible by individual studies only, but requires synthesis efforts where integrating data and results across multiple studies is crucial [Kö19, Ba19, Kü20, He21]. At the same time, great technological advances have led to ever increasing data output that must be managed in a multitude of ways [KMW20]. The domain of computer science can supply techniques to tackle these issues, however, uptake by non-domain experts is slow [Sa19], as with ever increasing complexity of technologies, it becomes more and more difficult for them to adopt new systems and leverage their upsides. Therefore, we aim to investigate ways to mitigate these challenges on an example use case and hope to contribute towards a solution. In this paper, we describe our efforts to tackle the interaction problem between interdisciplinary researchers and advanced semantic technologies to improve data synthesis in biodiversity research by supporting the creation of integrated, FAIR [Wi16] datasets that lead to an increase in research data quality, while reducing the manual effort needed for data collection, cleaning, and integration. We contribute by presenting a case study within the context of PlantHub[5], building a knowledge graph consisting of plant trait data from the TRY plant trait database [Ka11, Ka20], citizen science occurrence data from naturgucker.de [GB24b], and add taxonomic and additional information from multiple sources including Wikidata [VK14], GBIF [GB24a], IUCN [IU24], and others. We provide a query builder interface to simplify interaction with the knowledge graph, allowing queries to be asked by any user without prior knowledge of SPARQL[6] or the required vocabularies and ontologies. The query builder simplifies knowledge retrieval from the graph, compiles available data on the species level, links to sources, and allows for visualization of query results on a world map. Forthwith, we describe the background, tools, and workflows to construct the knowledge graph, showcase the query builder interface, and discuss the results.

2 Background and related work

Data acquisition and synthesis in biodiversity research. In the domain of ecology, researchers are met with complex problems. For a lot of them, it is insufficient or impossible to collect new research data in the field. In many of these cases, data can or must be synthesized from existing sources to perform new analyses on, potentially gaining new knowledge for the research field [LS21, He21, Or22]. This is a common approach when conducting complex meta analyses [KG13, Gu18, Pi20, Be23] and can also be valuable

[5] https://planthub.idiv.de/
[6] https://www.w3.org/TR/sparql11-query/

when evaluating one's field data. The resulting dataset should adhere to the FAIR principles of findability, accessibility, interoperability, and reusability. Common tasks in ecology are matching of species names and synonyms, enriching single species entries with additional information like taxonomies or traits, finding common, standardized representations for concepts, method descriptions, or appropriate modeling of geolocation. Completing these tasks is both costly in effort and time spent. We argue that knowledge graphs can simplify adherence to the FAIR principles and thus improve the potential for reuse in future synthesis.

Knowledge gaps in biodiversity research and integrated databases. Many studies have pointed to significant knowledge gaps and data deficiencies existing in biodiversity research [Mc20, Ho21a, Is23]. These gaps exist on a variety of dimensions including species information, geographic distribution, functional traits and ecological functions of species, ecological interactions, and more [Ho15]. In part, these gaps exist because studies collecting data to fill them have not been conducted yet, however, in many cases relevant studies and data exist, but access and discovery is difficult [St17, Ga21]. In identifying these knowledge and research gaps, it becomes crucial to ensure that already available data and studies are used such that no information is lost or collected repeatedly. Integrated data sources have been established as the state of the art solution to this problem [JMG12, Kö19, He21]. These integrated sources are large-scale databases serving a multitude of functions, however, they also differ in the types of information they represent and may use different taxonomies and as such, it has been shown that significant improvements of taxonomic coverage in distribution and trait data may be achieved by integrating further over these databases [Fe22]. Another problem is that manual, one-time integration might only have limited use since sources evolve over time. Therefore, solutions are required that allow data and sources to be kept up-to-date and present functionalities to make this process simpler. The research field of knowledge graph evolution proposes solutions to this issue [Bo19, Zh23].

PlantHub. In an effort to tackle these issues, the PlantHub project (planthub.idiv.de) was conceived at the German Centre for Integrative Biodiversity Research (iDiv). PlantHub is a web portal that aims to expose open-access, plant-related data in the biodiversity and ecology domain and increase interoperability among projects and with external databases. To this end, extensive data processing and management workflows are required for harmonization [Sc23]. Another priority of PlantHub is to enhance the visability of integrated data sources by presenting tools for exploration, visualization, and analysis.

Knowledge graphs and ontologies. In recent years, semantic technologies have been proposed as tools to support biodiversity research and improve data integration and synthesis next to integrated databases [Pa16, Pa19, Pe19, Ba21]. With ontologies as common vocabularies and schemata for organizing data, and knowledge graphs instantiating them with real world entities, it becomes possible to combine knowledge from multiple sources into one integrated knowledge base. Hence, as another advantage for data synthesis, it becomes possible to seamlessly communicate and fetch data between knowledge bases [Ho21b]. Through the reuse potential inherent in their design, knowledge graphs and ontologies make strides towards enhancing community-wide data quality and adherence to the FAIR principles. As a forerunner for already existing technology, Wikidata has been proposed as a knowledge graph for the life sciences with multiple, varying use cases [Wa20].

3 Method

In this paper, we present how semantic technologies can be applied in biodiversity research to improve data synthesis, integration, and quality. We first motivate our work with two use cases, then describe how and with which tools we build a knowledge graph from multiple sources and provide a query builder interface for the graph, and finally showcase the result.

3.1 Use Cases

Our work can be motivated by both short-term and long-term advantages of representing data as knowledge graphs. For the former, imagine a typical situation in the life of a PhD student in biodiversity: They went to the field and collected semi-structured data in a spreadsheet. This contains possibly identification for the object of the study, e.g. names of the species or other taxonomic ranks, and observations relating to those species. One goal could be to match said species to an integrated data source like GBIF to find accepted species names and compare them to possible synonyms, retrieve the species' taxonomy, or to find its respective GBIF ID, which is a unique identifier to any species and observation of a species in the database. With these identifiers, they could in a second step search other integrated data sources and knowledge graphs for identifier occurrences and enrich their data with outside knowledge. In our example, we use GBIF and its identifiers to match species and search for entries of them in Wikidata, through which multitudes of data can be sourced, including identifiers to other sources, properties, media, and more. Here, we retrieve other identifiers through Wikidata, especially that of IUCN [Co01], and are therefore able to enrich our original data with the IUCN Red List's[7] conservation status.
In the long-term, this dataset together with others could contribute to a meta-analysis. These analyses rely on the synthesis (or in computer science terms: integration) of numerous datasets. Today, this is a labor-intensive task, often requiring months if not years of PostDoc time. The more datasets are FAIR and available as knowledge graphs, the less effort will be involved in compiling them, and the quicker important research questions will be answered.

3.2 Constructing a knowledge graph

In this section, we describe the steps and tools used to build a knowledge graph from our data sources. We combine plant trait data from the TRY database with observational data from naturgucker.de, hosted on GBIF. We use Ontotext Refine[8] for multiple purposes, including data analysis, text transformation, execution of API calls and SPARQL queries to other sources, and RDF mapping. Considering that the data used in this work is already research grade, not much cleaning is needed. Therefore, we focus on more complex functionalities.
Using APIs and SPARQL queries. We use the OpenElevation API[9] to retrieve the elevation

[7] https://www.iucnredlist.org/
[8] https://www.ontotext.com/products/ontotext-refine/
[9] https://open-elevation.com/

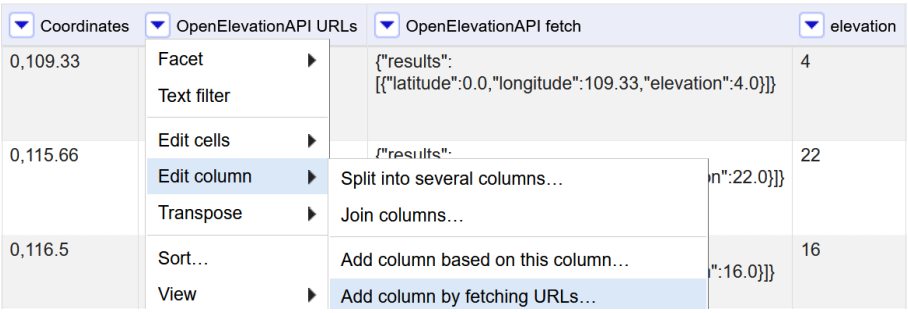


Fig. 1: Using APIs in Ontotext Refine.

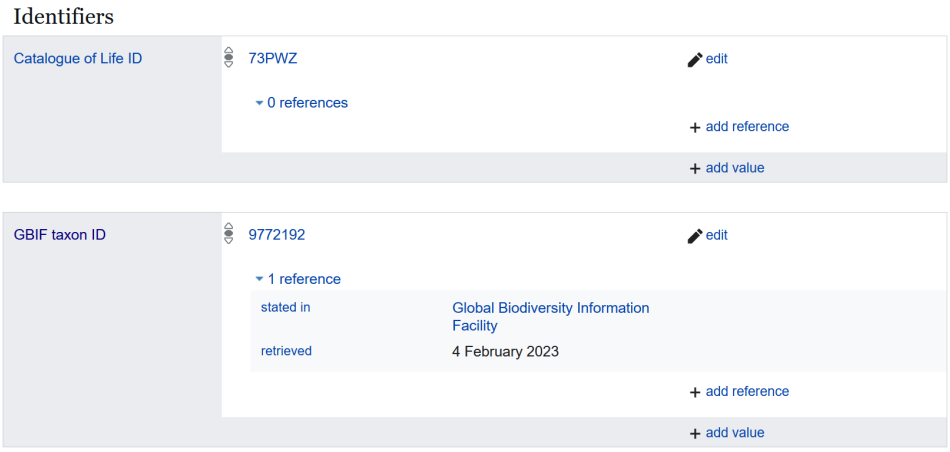


Fig. 2: Multiple identifiers of a species as listed on Wikidata.

of coordinates given in TRY entries. To this end, we combine the split coordinate cells into a new column. The values of this coordinate column may then be used to construct the URL strings needed for API access. These API calls can then be executed on every cell in a column and the result is returned to a target column, that can be accessed through JSON parsing and other transformations. These steps are also shown in Fig. 1. The process of API fetching is used not only for elevation, but also to retrieve additional information from GBIF[10], such as taxonomies or to compare identifiers and species names. We also use the Red List API[11] to retrieve a species' conservation status. The execution of SPARQL queries works analogous to that of API calls. We use SPARQL to query Wikidata for multiple identifiers and images attached to species entries. Fig. 2 shows how multiple identifiers of a species are listed on Wikidata. Using a GBIF identifier as input, we query Wikidata for an entity that has a property called *GBIF taxon ID*, with a value matching the input.

[10] https://techdocs.gbif.org/en/openapi/v1/species
[11] https://apiv3.iucnredlist.org/api/v3/docs

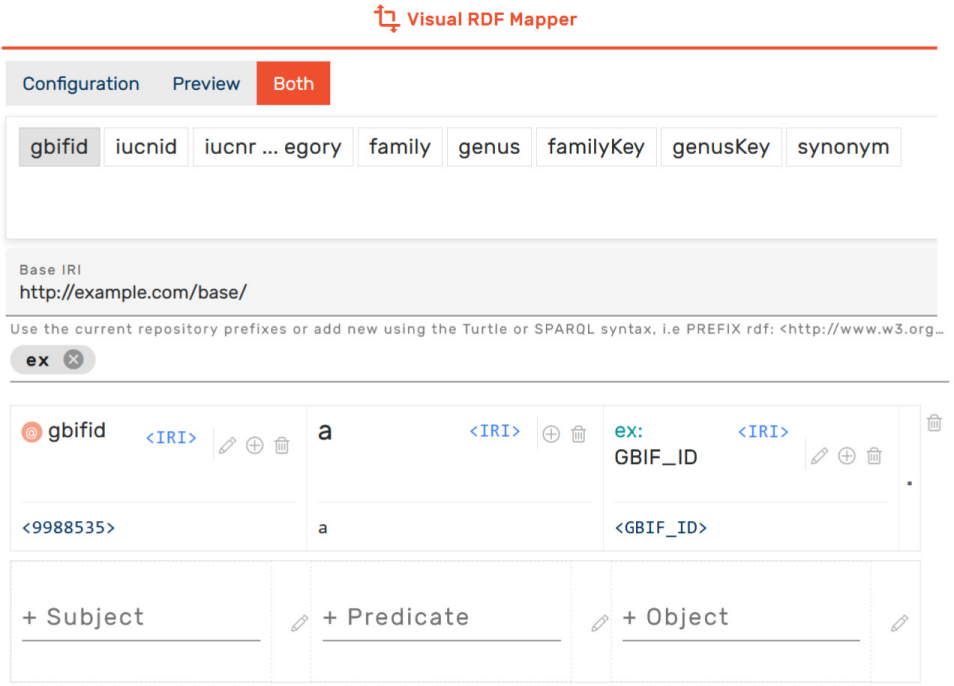


Fig. 3: Visual RDF Mapper in Ontotext Refine.

RDF mapping. One or multiple ontologies are needed as schemata for a knowledge graph, that is subsequently instantiated by the prepared tabular data. At the core of the PlantHub data are observations of plant species occurrences and traits. Therefore, we have chosen SOSA/SSN[12] as our main ontology as it aims to describe scientific observations (both sensor-based and human-made). With SOSA/SSN, it is possible to state the specific feature of an observation, which property was measured and with which procedure, multiple time properties such as result time, which sensors were used to make the observation, and deliver a description of the result. To instantiate chosen ontologies with tabular data, a mapping must be performed between table columns and ontology terms. The visual RDF mapper extension in Ontotext Refine, shown in Fig. 3 is used to create these mappings. With a visual tool, mapping becomes considerably easier when compared to traditional methods in which mappings are constructed using code.

Graph store. We use Ontotext GraphDB[13] to store the knowledge graph. GraphDB supplies many useful functionalities, including tools to explore and inspect the graph visually, show class hierarchies and relationships, as well as a SPARQL interface that allows queries to be performed on the knowledge graph. As such, it also acts as a SPARQL endpoint that can be

[12] https://www.w3.org/TR/vocab-ssn/
[13] https://www.ontotext.com/products/graphdb/

queried by other applications such as the query builder in this work.
Reproducibility. We provide a GitHub repository[14] containing the data used, the Ontotext Refine project files and operation histories, and the R2RML[15] mappings to create the graph.

3.3 Query builder showcase

To simplify interaction with the knowledge graph, we built an interface that allows users to construct complex SPARQL queries by filling input fields in a graphical interface embedded in the PlantHub website. In this section, we will show all functionalities of this interface.

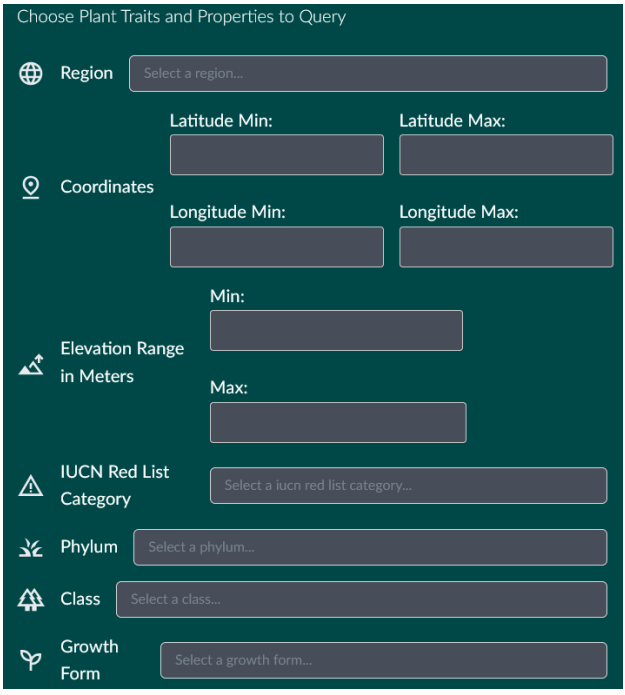


Fig. 4: Input fields of the query builder.

Showcase. The top of the query builder interface, as seen in Fig. 4, contains multiple input fields with dropdown menus for predefined options. Users can select multiple regions, or a range of coordinates and elevation spans to restrict results. Below that, users can filter for different plant traits or properties, or select specific taxonomic ranks to search. These properties include the IUCN category, growth form, leaf shape, leaf phenology type, and plant growth substrate, among others. Once selections have been made, the query can be

[14] https://github.com/fusion-jena/PlantHub-QueryBuilder
[15] https://www.w3.org/TR/r2rml/

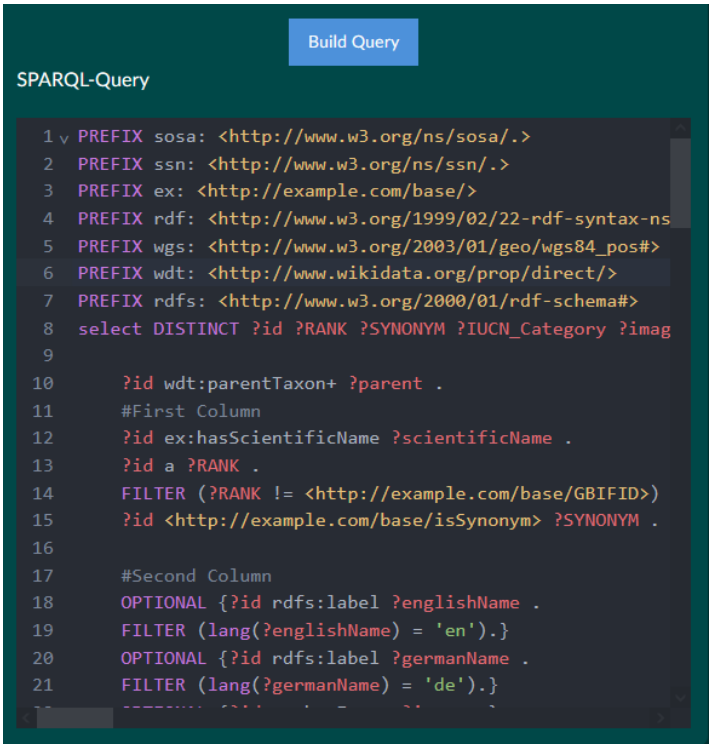


Fig. 5: Code editor showing the query.

built with the press of a button and shown in a code editor, as presented in Fig. 5, that experienced users may also use to formulate their own queries by hand. Once users are satisfied with the query, it can be sent to the graph and results are returned in a table below. In this table, shown in Fig. 6, all available information for a species is compiled to a single row. The collected information includes, if available, the scientific name and authors or the species, whether the name is accepted or a synonym, its IUCN category, multiple labels, an image sourced from Wikidata, identifiers and links to the corresponding pages of GBIF, Wikidata, IUCN, WFO[16] [Bo20], and IPNI[17] [Cr99]; any traits measured for the species, and its complete taxonomy, linked to the GBIF usage key of the entry on each taxonomic rank. To display all occurrences returned by the query, or all occurrences of a single species, we use OpenStreetMap[18], shown in Fig. 7. The color of the dots on the map corresponds to the elevation of the occurrence. When hovering over single dots, information about the sighting is displayed. This allows users to trace which plant trait was observed where.

[16] https://www.worldfloraonline.org/
[17] https://www.ipni.org/
[18] https://www.openstreetmap.org/about

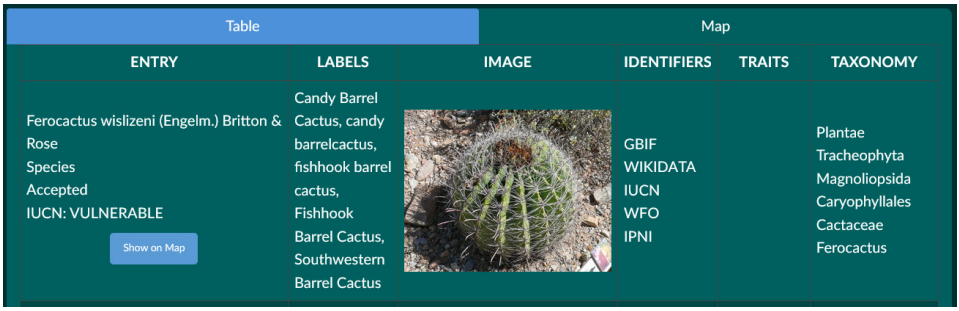


Fig. 6: Results are compiled at species level and shown in a table.

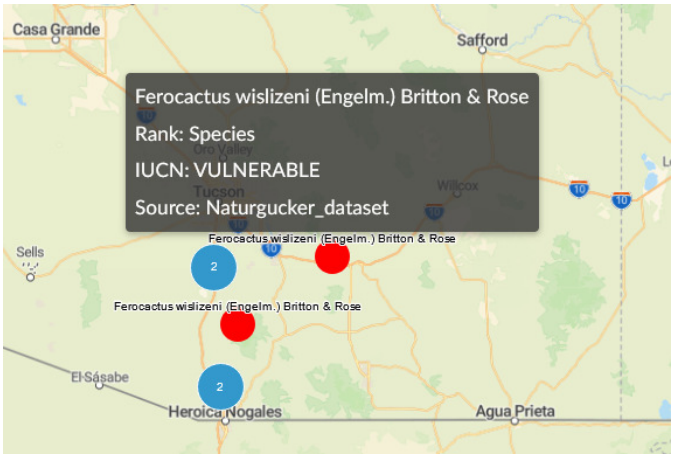


Fig. 7: Results for a single species shown on a map.

4 Discussion

The query builder interface presents users with options to search and explore the knowledge graph in a simplified manner. Through multiple query choices and varied representation of results, we aim to bridge the gap between knowledge graphs and ontologies as theoretical concepts and technologies from computer science, to how they can be applied in practice in the domain of ecology and biodiversity research. Hopefully, the potential of knowledge graphs, especially with their strength of seamless communication between sources and the advantage of inherent adherence to FAIR principles, has become more clear in this work. However, this has only been a first step in communicating and bringing semantic technologies closer to practice. There are many open problems and opportunities in this field of research. What additional use cases for the use of semantic technologies in these domains exist and how can we better communicate untapped benefits? How can construction of new knowledge graphs and the search and implementation of existing ontologies be simplified?

What are challenges and approaches in keeping existing knowledge graphs up-to-date? How can we as computer scientists deliver actionable solutions to the interaction problem? We aim to explore the great potential that large language models have in solving some of these issues. How can LLMs be used as a natural language interface for knowledge graph usage? Can strict adherence of LLMs to ontologies be forced, therefore enhancing model outputs? At which points in the knowledge graph construction and ontology mapping process may they be used as tools? There is a vast variety of open questions in this field of research, some of which we aim to answer in our future work. In the context of a new project at iDiv, iAnswer, we aim to contribute to solutions to the problem of knowledge graph creation and usage in biodiversity and ecology research, applied to iDiv data sources.

5 Conclusion

In this paper, we discussed the challenges for data synthesis and integration in ecology and biodiversity research. Though a wide variety of integrated data sources already exist in the domain, manual synthesis and integration continue to present challenging and time-intensive tasks like species name matching, synonym resolution, enriching entries with taxonomies, traits, or distribution ranges, etc., as well as open problems like communication and integration between mentioned sources. We posit that knowledge graphs and ontologies have great potential in aiding these processes and may improve research practice, though their application has gained little traction so far. The biggest reason for this appears to be the vast domain knowledge required for ontology modeling and knowledge graph construction and usage, as well as a high degree of specificity required from ontologies. Therein lies our motivation for this work, presenting a knowledge graph consisting of plant trait data hosted on PlantHub and plant occurrence data from naturgucker.de and a query builder interface as an attempt to alleviate the interaction problem by allowing SPARQL queries to be formulated and sent to the graph without domain knowledge and displaying the results in a compiled manner and on a world map. In future work, we will explore how large language models may be leveraged to improve knowledge graph creation and interaction for non-domain experts in the context of a new iDiv project, iAnswer.

Acknowledgments

We gratefully acknowledge the support of the German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG–FZT 118, 202548816).

We thank Florian Hähnert, Cedric Julian Wittkuhn, Onur Yazicioglu, and Ali Yavarnegoon for their initial work on the query builder interface that started as a student project.

Bibliography

[Ba19] Ball-Damerow, Joan E; Brenskelle, Laura; Barve, Narayani; Soltis, Pamela S; Sierwald, Petra; Bieler, Rüdiger; LaFrance, Raphael; Arino, Arturo H; Guralnick, Robert P: Research applications of primary biodiversity databases in the digital age. PloS one, 14(9):e0215794, 2019.

[Ba21] Babalou, Samira; Schellenberger Costa, David; Kattge, Jens; Römermann, Christine; König-Ries, Birgitta: Towards a semantic toolbox for reproducible knowledge graph generation in the biodiversity domain-how to make the most out of biodiversity data. In: INFORMATIK 2021. Gesellschaft für Informatik, Bonn, pp. 581–590, 2021.

[Be23] Beaumelle, Léa; Tison, Léa; Eisenhauer, Nico; Hines, Jes; Malladi, Sandhya; Pelosi, Céline; Thouvenot, Lise; Phillips, Helen RP: Pesticide effects on soil fauna communities—a meta-analysis. Journal of Applied Ecology, 60(7):1239–1253, 2023.

[Bo19] Bonatti, Piero Andrea; Decker, Stefan; Polleres, Axel; Presutti, Valentina: Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371). In: Dagstuhl reports. volume 8. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[Bo20] Borsch, Thomas; Berendsohn, Walter; Dalcin, Eduardo; Delmas, Maïté; Demissew, Sebsebe; Elliott, Alan; Fritsch, Peter; Fuchs, Anne; Geltman, Dmitry; Güner, Adil et al.: World Flora Online: Placing taxonomists at the heart of a definitive and comprehensive global resource on the world's plants. Taxon, 69(6):1311–1341, 2020.

[Co01] Commission, Natural Resources. Species Survival: IUCN Red List categories and criteria. IUCN, 2001.

[Cr99] Croft, J; Cross, N; Hinchcliffe, S; Lughadha, E Nic; Stevens, PF; West, JG; Whitbread, G: Plant names for the 21st century: the International Plant Names Index, a distributed data source of general accessibility. Taxon, 48(2):317–324, 1999.

[Fe22] Feng, Xiao; Enquist, Brian J; Park, Daniel S; Boyle, Brad; Breshears, David D; Gallagher, Rachael V; Lien, Aaron; Newman, Erica A; Burger, Joseph R; Maitner, Brian S et al.: A review of the heterogeneous landscape of biodiversity databases: Opportunities and challenges for a synthesized biodiversity knowledge base. Global Ecology and Biogeography, 31(7):1242–1260, 2022.

[Ga21] Gadelha Jr, Luiz MR; de Siracusa, Pedro C; Dalcin, Eduardo Couto; da Silva, Luis Alexandre Estevao; Augusto, Douglas A; Krempser, Eduardo; Affe, Helen Michelle; Costa, Raquel Lopes; Mondelli, Maria Luiza; Meirelles, Pedro Milet et al.: A survey of biodiversity informatics: Concepts, practices, and challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11(1):e1394, 2021.

[GB24a] GBIF: The Global Biodiversity Information Facility: What is GBIF?, 2024. Accessed: 29 November 2024.

[GB24b] GBIF.Org (14 March 2024): Occurrence Download https://doi.org/10.15468/dl.ku5qss, 2024.

[Gu18] Gurevitch, Jessica; Koricheva, Julia; Nakagawa, Shinichi; Stewart, Gavin: Meta-analysis and the science of research synthesis. Nature, 555(7695):175–182, 2018.

[Ha22] Habibullah, Muzafar Shah; Din, Badariah Haji; Tan, Siow-Hooi; Zahid, Hasan: Impact of climate change on biodiversity loss: global evidence. Environmental Science and Pollution Research, 29(1):1073–1086, 2022.

[He21] Heberling, J Mason; Miller, Joseph T; Noesgaard, Daniel; Weingart, Scott B; Schigel, Dmitry: Data integration enables global biodiversity synthesis. Proceedings of the National Academy of Sciences, 118(6):e2018093118, 2021.

[Ho15] Hortal, Joaquín; de Bello, Francesco; Diniz-Filho, José Alexandre F; Lewinsohn, Thomas M; Lobo, Jorge M; Ladle, Richard J: Seven shortfalls that beset large-scale knowledge of biodiversity. Annual review of ecology, evolution, and systematics, 46(1):523–549, 2015.

[Ho21a] Hochkirch, Axel; Samways, Michael J; Gerlach, Justin; Böhm, Monika; Williams, Paul; Cardoso, Pedro; Cumberlidge, Neil; Stephenson, Peter J; Seddon, Mary B; Clausnitzer, Viola et al.: A strategy for the next decade to address data deficiency in neglected biodiversity. Conservation Biology, 35(2):502–509, 2021.

[Ho21b] Hogan, Aidan; Blomqvist, Eva; Cochez, Michael; d'Amato, Claudia; Melo, Gerard De; Gutierrez, Claudio; Kirrane, Sabrina; Gayo, José Emilio Labra; Navigli, Roberto; Neumaier, Sebastian et al.: Knowledge graphs. ACM Computing Surveys (Csur), 54(4):1–37, 2021.

[Is23] Isbell, Forest; Balvanera, Patricia; Mori, Akira S; He, Jin-Sheng; Bullock, James M; Regmi, Ganga Ram; Seabloom, Eric W; Ferrier, Simon; Sala, Osvaldo E; Guerrero-Ramírez, Nathaly R et al.: Expert perspectives on global biodiversity loss and its drivers and impacts on people. Frontiers in Ecology and the Environment, 21(2):94–103, 2023.

[IU24] IUCN: The IUCN Red List of Threatened Species. Version 2024-2, 2024. Accessed: 29 November 2024.

[JMG12] Jetz, Walter; McPherson, Jana M; Guralnick, Robert P: Integrating biodiversity distribution knowledge: toward a global map of life. Trends in ecology & evolution, 27(3):151–159, 2012.

[Ka11] Kattge, Jens; Diaz, Sandra; Lavorel, Sandra; Prentice, I Colin; Leadley, Paul; Bönisch, Gerhard; Garnier, Eric; Westoby, Mark; Reich, Peter B; Wright, Ian J et al.: TRY–a global database of plant traits. Global change biology, 17(9):2905–2935, 2011.

[Ka20] Kattge, Jens; Bönisch, Gerhard; Díaz, Sandra; Lavorel, Sandra; Prentice, Iain Colin; Leadley, Paul; Tautenhahn, Susanne; Werner, Gijsbert DA; Aakala, Tuomas; Abedi, Mehdi et al.: TRY plant trait database–enhanced coverage and open access. Global change biology, 26(1):119–188, 2020.

[KG13] Koricheva, Julia; Gurevitch, Jessica: Place of meta-analysis among other methods of research synthesis. Handbook of meta-analysis in ecology and evolution, pp. 3–13, 2013.

[KMW20] Kays, Roland; McShea, William J; Wikelski, Martin: Born-digital biodiversity data: Millions and billions. Diversity and Distributions, 26(5):644–648, 2020.

[Kö19] König, Christian; Weigelt, Patrick; Schrader, Julian; Taylor, Amanda; Kattge, Jens; Kreft, Holger: Biodiversity data integration—the significance of data resolution and domain. PLoS biology, 17(3):e3000183, 2019.

[Kü20] Kühl, Hjalmar S; Bowler, Diana E; Bösch, Lukas; Bruelheide, Helge; Dauber, Jens; Eichenberg, David; Eisenhauer, Nico; Fernández, Néstor; Guerra, Carlos A; Henle, Klaus et al.: Effective biodiversity monitoring needs a culture of integration. One Earth, 3(4):462–474, 2020.

[LS21] Ladouceur, Emma; Shackelford, Nancy: The power of data synthesis to shape the future of the restoration community and capacity. Restoration Ecology, 29(1):e13251, 2021.

[Mc20] McDonald, Robert I; Mansur, Andressa V; Ascensão, Fernando; Colbert, M'lisa; Crossman, Katie; Elmqvist, Thomas; Gonzalez, Andrew; Güneralp, Burak; Haase, Dagmar; Hamann, Maike et al.: Research gaps in knowledge of the impact of urban growth on biodiversity. Nature Sustainability, 3(1):16–24, 2020.

[Or22] Orr, Michael C; Hughes, Alice C; Costello, Mark J; Qiao, Huijie: Biodiversity data synthesis is critical for realizing a functional post-2020 framework. Biological Conservation, 274:109735, 2022.

[Pa16] Page, Roderic: Towards a biodiversity knowledge graph. Research Ideas and Outcomes, 2, 2016.

[Pa19] Page, Roderic: Ozymandias: a biodiversity knowledge graph. PeerJ, 7:e6739, 2019.

[Pe19] Penev, Lyubomir; Dimitrova, Mariya; Senderov, Viktor; Zhelezov, Georgi; Georgiev, Teodor; Stoev, Pavel; Simov, Kiril: OpenBiodiv: a knowledge graph for literature-extracted linked open data in biodiversity science. Publications, 7(2):38, 2019.

[Pi20] Pilotto, Francesca; Kühn, Ingolf; Adrian, Rita; Alber, Renate; Alignier, Audrey; Andrews, Christopher; Bäck, Jaana; Barbaro, Luc; Beaumont, Deborah; Beenaerts, Natalie et al.: Meta-analysis of multidecadal biodiversity trends in Europe. Nature communications, 11(1):3486, 2020.

[Sa19] Sachs, Joel; Page, Roderic; Baskauf, Steven J; Pender, Jocelyn; Lujan-Toro, Beatriz; Macklin, James; Comspon, Zacchaeus: Training and hackathon on building biodiversity knowledge graphs. Research Ideas and Outcomes, 5:e36152, 2019.

[Sc23] Schellenberger Costa, David; Boehnisch, Gerhard; Freiberg, Martin; Govaerts, Rafaël; Grenié, Matthias; Hassler, Michael; Kattge, Jens; Muellner-Riehl, Alexandra N; Rojas Andrés, Blanca M; Winter, Marten et al.: The big four of plant taxonomy–a comparison of global checklists of vascular plant names. New Phytologist, 240(4):1687–1702, 2023.

[Sh22] Shivanna, Kundaranahalli R: Climate change and its impact on biodiversity and human welfare. Proceedings of the Indian National Science Academy, 88(2):160–171, 2022.

[SSS21] Singh, Vijeta; Shukla, Shikha; Singh, Anamika: The principal factors responsible for biodiversity loss. Open Journal of Plant Science, 6(1):011–014, 2021.

[St17] Stephenson, PJ; Bowles-Newark, Nadine; Regan, Eugenie; Stanwell-Smith, Damon; Diagana, Mallé; Höft, Robert; Abarchi, Harouna; Abrahamse, Tanya; Akello, Christine; Allison, Hilary et al.: Unblocking the flow of biodiversity data for decision-making in Africa. Biological Conservation, 213:335–340, 2017.

[VK14] Vrandečić, Denny; Krötzsch, Markus: Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10):78–85, 2014.

[Wa20] Waagmeester, Andra; Stupp, Gregory; Burgstaller-Muehlbacher, Sebastian; Good, Benjamin M; Griffith, Malachi; Griffith, Obi L; Hanspers, Kristina; Hermjakob, Henning; Hudson, Toby S; Hybiske, Kevin et al.: Wikidata as a knowledge graph for the life sciences. Elife, 9:e52614, 2020.

[Wi16] Wilkinson, Mark D; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Bonino; Bourne, Philip E et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific data, 3(1):1–9, 2016.

[Zh23] Zhong, Lingfeng; Wu, Jia; Li, Qian; Peng, Hao; Wu, Xindong: A comprehensive survey on automatic knowledge graph construction. ACM Computing Surveys, 56(4):1–62, 2023.

Historic to FAIR: Leveraging LLM for Historic Term Identification and Standardization

Jan Fillies[1,2], Maximilian Teich[3], Naouel Karam[1], Adrian Paschke[1,2,4] und Malte Rehbein[3]

Abstract: As the availability of historical biodiversity data continues to grow, ensuring its usability through adherence to FAIR principles (Findable, Accessible, Interoperable, and Reusable) has become increasingly essential. This study addresses a key challenge in biodiversity data drawn from historical texts: identifying and interpreting common species names and scientific names. We highlight five main issues associated with historical common names: variations in spelling, the creation of new terms, shifts from broad historical names to more specific modern ones (and vice versa), and the renaming of historical terms. To tackle these challenges, we explore the application of a large language model (GPT-4) for entity detection and terminology alignment. Our findings demonstrate that GPT-4, when provided with a small context, can effectively identify both historical common species names and modern scientific names.

On a test dataset, the model achieved a 92% success rate in detecting historical common names and correctly identified 98% of scientific terms. Additionally, for four of the five identified challenges, the LLM provided meaningful insights, including successfully matching historical common names to their modern counterparts. We demonstrate an embedded understanding of the evolution of biodiversity terminology within the model which underscores its potential to mobilize historical biodiversity data according to FAIR.

Keywords: Large Language Models, FAIR Principals, Language Standardization, Data Interoperability, Historic Data, Semantic Annotation, Taxonomies

1 Introduction

Human observations and reflections on nature have been documented for centuries. These historical records, when properly processed, serve as invaluable data for modern science, particularly in enhancing biodiversity datasets to be more Findable, Accessible, Interoperable, and Reusable (FAIR). However, the evolving language used to describe species, concepts, and ecological relationships presents a significant challenge. Terminological, linguistic, and conceptual shifts over time must be preserved and understood for a deeper analysis [Ko21].

As biodiversity research advances, aligning historical data with standardized taxonomies, ontologies, and authority files has become increasingly crucial for supporting FAIR principles. Controlled taxonomies ensure a consistent and uniform vocabulary within the field [Sv00]. Yet, the evolution of language remains a persistent hurdle. From a scientific

[1] Institute for Applied Informatics (InfAI), University of Leipzig, Germany,
[2] Freie Universität Berlin, Berlin, Germany,
[3] Chair of Computational Humanities, University of Passau, Germany,
[4] Fraunhofer FOKUS, Berlin, Germany

doi:10.18420/BTW2025-121

perspective, species are frequently reclassified, merged, split, or renamed due to new discoveries or paradigm shifts [RG14]. Societal influences further complicate matters, with common names varying regionally, changing over time, or becoming obsolete altogether. The issue is compounded by the lack of standardized nomenclature in historical biodiversity records, where only local common names were often recorded. Making these datasets accessible and FAIR-compliant requires manually matching historical common names with modern taxonomies. This process demands deep expertise in both contemporary biodiversity classification and historical linguistic variations, creating high barriers to consistency and usability in integrating historical data into modern studies.

This study focuses on identifying historical common names from a biodiversity dataset harvested from records of Württemberg between 1824 and 1886 and matching them to modern scientific equivalents within a standardized taxonomy. Recent advancements in large language models (LLMs) have shown promise in biodiversity research [Os23], with models like GPT-3.5 demonstrating utility in species descriptions, occurrences, and taxonomies [EF23]. LLMs have also proven effective in entity recognition tasks [AL23]. Unlike traditional NER approaches, which require labeled training data that is usually unavailable for historical biodiversity texts, LLMs can generalize from their pretraining to recognize and infer entity mappings even in the absence of explicit training examples. Based on the positive findings for [AL23], we evaluate the capabilities of GPT-4o to identify historical common and scientific names, align them with modern taxonomy, and address key challenges in processing historical biodiversity data.

By examining the model's ability to interpret historical language within a scientific context, this research contributes to the growing field of computational approaches for mobilizing historical biodiversity data. As digitization efforts expand (e.g., [Go24, Ba28, Vi22]), the methods and findings presented here hold significant potential for enhancing the usability and consistency of these historical datasets.

This research is closely linked to the *BiodivPortal* [Ka24], an ontology repository and service aimed at unifying access, development, and maintenance of terminologies in biodiversity and environmental sciences. By leveraging *BiodivPortal* terminologies, the goal is to create a tool for semantic annotation of historical data that overcomes linguistic and conceptual shifts in biodiversity vocabulary over time. Aligning historical data with modern standards enhances accessibility, utility, and integration, supporting advancements in biodiversity research, environmental studies, and FAIR data principles.

In experiments, GPT-4o achieved high accuracy in identifying historical names: 92% for common names and 98% for scientific names when provided minimal context. Without context, accuracy dropped slightly to 91% and 96%. The study identified four key challenges in processing historical species names:

1. Variations in spelling;
2. Emergence of new terms;

3. Evolution from broad historical common names to specific modern equivalents;
4. Translation from modern to specific historical names; and
5. Renaming of historical common names.

The findings demonstrate that GPT-4o effectively addresses all challenges with offering direct solutions to four of them. This suggests that LLMs can significantly enhance human-guided efforts in standardizing historical biodiversity datasets, streamlining data integration processes, and reducing manual annotation time. While some challenges remain, LLMs show promising potential for modernizing and harmonizing biodiversity records.

The paper is organized as follows: Section 2 details the dataset used, Section 3 explains the applied prompting strategies, and Section 4 presents the experimental design and its results. Section 5 discusses implications and limitations of the findings, and Section 6 concludes and offers future research directions in. The produced code is available on GitHub[5].

2 Dataset

This research utilizes a dataset of animal names derived from 19th-century regional studies on the districts (*Oberämter*) of the southern-German kingdom of Württemberg. In 1820, the Royal Statistical Office (*Königl. Statistisch-Topographisches Bureau*) was established to systematically survey the kingdom's geography, history, culture, and ecology. Over several decades, the office distributed questionnaires and interviewed local experts, compiling detailed information from each district.

Between 1824 and 1886, the office published 64 volumes of the so-called *Oberamtsbeschreibungen*, each dedicated to one of the kingdom's districts and typically released at a pace of about one per year. A notable feature of these volumes is the inclusion of a chapter on the fauna of the respective district, providing valuable historical insights into regional biodiversity. These chapters serve as the foundation for the dataset used in this study, offering a rich source of information to explore the evolution of animal nomenclature and align historical data with modern taxonomic standards.

The extent of the fauna chapters in the *Oberamtsbeschreibungen* varies greatly. While some chapters merely state that no special or remarkable animals are present in a district, others provide detailed accounts, listing hundreds of species and discussing their significance to the local economy and culture. The lengths of these chapters range from a brief 18 tokens in the rural district of Künzelsau to a comprehensive 9,682 tokens in the well-documented university town of Tübingen, with a mean average chapter length of 885 tokens.

To create a gold-standard dataset, the animal names recorded in these chapters were manually extracted. This dataset includes both vernacular and scientific names as they appeared in the

[5] https://github.com/biodivportal/Historic_to_FAIR

historical texts, alongside the inferred presence of each species as a boolean value. Notably, important animals, particularly large mammals like deer, boars, and bears, were frequently mentioned as either absent or extinct.

The extraction process involved splitting the fauna chapters into 3,351 overlapping text chunks, each 200 tokens long. From a sample of 50 of these chunks, vernacular and scientific names were systematically extracted. Unless explicitly stated otherwise, animals mentioned in the text were assumed to have been present in the district. During the extraction process, several challenges emerged in matching historical names to their modern counterparts:

1. **Correction of historical spellings**: LLM outputs may attempt to modernise historical spellings to align with contemporary conventions. For example, the common kestrel might appear in 19th-century texts as "Thurmfalke" rather than the modern "Turmfalke."
2. **Creation of new words**: Historical texts sometimes use adjectives as part of animal names, which can lead to variations. For instance, the name "Wildschwein" (wild boar) might appear as "wildes Schwein" (feral pig) in historical records.
3. **General to specific**: Historical texts may refer to broader concepts that are now more specific in modern language, such as "Holzwespen" (wood wasps), which in modern terms may correspond to specific species.
4. **Specific to general**: Conversely, historical texts might use more specific terms that have since become generalized in modern usage, such as "Schwarzamsel" (black blackbird) being replaced by the broader term "Amsel" (blackbird).
5. **Renaming and dialectal variations**: Historical records often include local dialects or outdated terms for species. For instance, "Steißfuß" has been replaced by the modern "Haubentaucher" (great crested grebe), and "Feldhuhn" is now known as "Rebhuhn" (grey partridge).

3 Prompting

This research addresses two primary challenges: first, identifying all species mentioned in historical texts, and second, aligning these historical biodiversity terms with their modern-day equivalents. To tackle these challenges, the performance of the GPT-4o language model [Op24] was evaluated. GPT-4o was selected for its regular updates, substantial model size, and robust performance across a range of academic benchmarks [Mi24].

Building on the framework established by [Ek23], which categorizes prompt engineering into basic and advanced methodologies, this study focuses on the basic approach. This involves techniques such as control codes, templates, iterative testing, and refinement, prioritizing rapid testing and prototype development. Advanced strategies, such as temperature and token control, prompt chaining, and dynamic prompt adaptation, are better suited for fine-tuning

systems in more developed stages. Given the exploratory nature of this study, we employed standard templates with an emphasis on iterative refinement to achieve maximum accuracy.

To identify historical common names, and to align them with modern taxonomies, two prompts were developed. Both prompts share a consistent structure: a brief contextual introduction, a clear task description, and specifications for the expected output format.

For example, the prompt designed to identify historical common names reads: “Identify all common names for species in this text: {text} Please think deeply and provide JUST a list comma separated list, no numbers, include of all named common name species. ADD NOTHING provide just the list."

The prompts were further refined to test whether providing additional contextual information improved the model’s predictions. In these cases, the prompt was expanded to include a historical setting, such as: “You are a biodiversity expert. Identify all common names for species in this an old German (Württemberg) text: {text}."

In the second phase of the experiment, the model was tasked with aligning historical common names to their modern-day counterparts and proving a brief explanation for each match. This phase aimed to evaluate the model’s ability to contextualize and translate historical terminology into contemporary equivalents. The adjusted prompt reads: “You are a biodiversity expert. If necessary, provide the modern common names for these historic common names found in an old German (Württemberg) text, provide short explanations.”

4 Experimental Results

The first part of the experiment evaluated GPT-4o’s ability to detect historic common names and historic scientific names within the provided text. A sample of the results for identifying common names is displayed in Table 1, while results for identifying scientific names are presented in Appendix A, Table 8. A human annotator reviewed and evaluated all results. Small deviation in the prediction result was deemed acceptable (e.g., “Gnophos furvata” instead of “Gnophos furvata S. V.”). The complete set of results is available on GitHub[6].

Table 2 summarizes the overall performance in identifying common names and scientific names from the historical dataset. The data shows that most (91% accuracy) common names were correctly identified without additional context. A slight increase (to 92% accuracy) is seen with the provision of situational context. The model’s performance in identifying scientific names was even higher, achieving 96% accuracy without additional context and improving to 98% when situational context was provided.

During the identification of common names, five primary sources of error were identified:

[6] https://github.com/biodivportal/Historic_to_FAIR

Tab. 1: Extract of LLM based Identification of Historic German Common Species Names

Provided Text	LLM Prediction (Context)	LLM Prediction (No Context)	Annotated Common Names
An Fischen ist der Bezirk ebenfalls arm. In der Glatt findet sich die Bachforelle (Salmo Fario) und der Schuppfisch (Ciprinus Dobula), im Neckar außer letzterem auch die Barbe (Cyprinus Barbus), die Nase, gewöhnlich Weißfisch genannt (C. Nasus), die Groppe (Cottus Gobio) und der Aal (Muraena Anguilla).	Bachforelle, Schuppfisch, Barbe, Nase, Weißfisch, Groppe, Aal, Flußkrebs, Geitze	Bachforelle, Schuppfisch, Barbe, Weißfisch, Groppe, Aal, Flußkrebs, Geitze	Bachforelle, Schuppfisch, Barbe, Nase, Weißfisch, Groppe, Aal, Flußkrebs, Geitze
Die Fauna des Bezirks hat manches Eigenthümliche und zeichnet sich vor den dem Schwarzwalde nicht angehörigen Theilen des Landes besonders aus. Von den Säugethieren des Waldes sind das wilde Schwein, welches früher in den weitgedehnten Waldungen so heimisch war und das Edelwild jetzt ganz verschwunden; viel länger schon ist dies der Fall bei den Bären, auf welche hier noch im 15. Jahrhundert gejagt wurde (Grimm, Weisthümer I, 384. 388).	wilde Schwein, Edelwild, Bären	wilde Schwein, Edelwild, Bären	wildes Schwein, Edelwild, Bär
Molytes dirus Hbst. auf dem Schloßberg; Grypedius equiseti Fabr. nicht selten an unserm größten Schachtelhalm auf einer feuchten Wiese unter Lustnau. – Daß Arten von Bostrychus in hiesiger Gegend durch ihre Menge besonderen Schaden den Waldbäumen zugefügt hätten, ist nicht bekannt geworden; wohl aber haben wir selber Äpfelbäume durch Eccoptogaster pruni Rtz. kranken sehen. – Von Bockkäfern mag erwähnt sein: Mesosa nebulosa Fabr.	Schloßberg, Schachtelhalm, Waldbäumen, Äpfelbäume	largest horsetail, apple trees	Bockkäfer

Tab. 2: Evaluation of Identification Performance in rounded Percent

LLM Prediction	Accuracy
Common Names no Context	0.91
Common Names with Context	0.92
Scientific Names no Context	0.96
Scientific Names with Context	0.98

1. **Selection of scientific names instead of common names**: The model occasionally returned scientific names rather than the requested common names.
2. **Omission of terms**: Certain common names were omitted without a clear explanation for the omission.
3. **Failure to produce output**: On rare occasions, the model failed to generate any output at all.
4. **Language switching**: The model unexpectedly switched to English instead of remaining in the intended language of the historical text.
5. **Misidentification of landscape terms**: The model sometimes identified terms describing the landscape rather than species, as demonstrated in Table 1, row 3.

The primary source of error in the identification of scientific names was the omission of certain terms, which is not explainable from the outside. This underscores a potential limitation in the model's consistency when handling historical biodiversity data. The error regarding "Misidentification of landscape terms" could, to some extent, be caused by the prompt referring to "species," which also includes plant-based species and not just animals, as in the human annotation.

For the second phase of the experiment, tables 4 through 7 show manually selected examples representing each category of challenges outlined in Section 2. The model's performance in addressing these challenges was assessed through expert evaluation. Table 3 demonstrates that the model accurately translates historical spellings into their modern equivalents for common names. Table 4 highlights the model's ability to identify newly formed species names that combine an adjective with a species name. Table 5 shows that the model effectively refines general historical common names into more specific modern equivalents. Table 6 illustrates the model's capacity to generalize specific historical terms into broader modern common names. Table 7 reveals that even in cases where words and their semantic meanings have significantly shifted over time, the model successfully identifies the correct modern equivalents. Apart from a single error in mapping a historical general concept to a specific modern-day counterpart, see Table 5, the model achieved an accuracy of 66.33% across the three statements. All other examples were mapped correctly, achieving 100% accuracy in their categories.

5 Discussion and Limitations

The findings show that GPT-4o can effectively extract both common modern names and modern scientific names from historical texts, even without fine-tuning. While the ability to recognize names might appear unremarkable given the efficiency of LLM-based entity recognition in various contexts, its extension to historical texts is notable and warrants further evaluation on larger datasets. Contextual information was found to marginally improve prediction accuracy, which is reasonable given the already high baseline performance

Tab. 3: Evaluation of the LLM's ability to correct historical spellings.

Common name	LLM Prediction	Human Evaluation
"Kukuk" (cuckoo)	Corresponds to "Kuckuck" today.	Correct
"Punktirte Wasserläufer" (spotted sandpiper)	Now "Punktierter Wasserläufer."	Correct
"Thurmfalke" (common kestrel)	Now "Turmfalke."	Correct
"Kriekente" (teal)	Modern terms are "Krickente."	Correct

Tab. 4: Evaluation of the LLM's ability to detect the creation of a new word.

Common name	LLM Prediction	Human Evaluation
"Wilde Ente" (wild duck)	Modern terms are "Wildente."	Correct
"dreizehige Specht" (three-toed woodpecker)	Modern names are "Dreizehenspecht."	Correct
"Wilde Schwein" (wild swine)	Modern term is "Wildschwein".	Correct

without context. However, some prediction errors proved inconsistent and non-reproducible, highlighting a limitation of GPT-4o as a proprietary model. The lack of transparency regarding its training data and underlying mechanisms hinders deeper investigation. Open-source models offer greater transparency and controllability for addressing such issues.

In the second phase of development, the model showed its capability in addressing common challenges in historical biodiversity data, including spelling variations and semantic shifts. It performed well in handling challenges 1, 2, 4, and 5, but struggled with challenge 3: translating broad historical concepts into specific modern-day counterparts. This difficulty is explainable, as identifying precise modern equivalents for broad historical terms is challenging even for many humans. Although the model could not directly solve this issue, it demonstrated the ability to provide relevant suggestions. For example, for the term "Edelwild" (noble game), the model suggested: "This term is rarely used today. It historically referred to higher-value game animals, specifically: "Rotwild" (red deer), "Damwild" (fallow deer), "Rehwild" (roe deer)."

The identification of modern-day common names shows great potential for meeting future FAIR data requirements. By aligning historical names with modern equivalents, the model demonstrates that LLMs can be used effectively for language standardization. This approach

Tab. 5: Evaluation of the LLM's ability to detect that general terms are more specific today.

Common name	LLM Prediction	Human Evaluation
"Holzwespen" (wood wasps)	A general family name, less specific compared to modern terms.	Correct
"Gemeine Möve" (common gull)	Likely refers to the "Silbermöwe" (herring gull) or another common gull species.	Not Correct more likely the Larus canus ("Sturmmöve", common gull)
"Edelwild" (noble game)	This term is rarely used today. It historically referred to higher-value game animals, specifically: "Rotwild" (red deer), "Damwild" (fallow deer), "Rehwild" (roe deer).	Correct
"Borkenkäfer" (bark beetle)	Still in use, though modern terms specify species more clearly.	Correct

Tab. 6: Evaluation of the LLM's ability to detect that specific terms are more general today.

Common name	LLM Prediction	Human Evaluation
Schwarzamsel (black blackbird)	Generally referred to simply as "Amsel" (blackbird) today.	Correct
Rothbraune Waldmaus (reddish-brown forest mouse)	Typically just called "Waldmaus" (wood mouse) today.	Correct

Tab. 7: Evaluation of the LLM's ability to detect that a species was renamed or in a local dialect.

Common name	LLM Prediction	Human Evaluation
"Feldhuhn" (field hen)	Likely corresponds to "Rebhuhn" (partridge) today.	Correct
"Holztaube" (wood pigeon)	Now typically called "Hohltaube" (stock dove) or "Ringeltaube" (wood pigeon) depending on the species.	Correct
"Gehaubte Steißfuß" (crested grebe)	Likely "Haubentaucher" (great crested grebe) and other grebe species.	Correct
"Groppe" (bullhead)	Still known but less commonly used today; also called "Mühlkoppe" in some regions.	Correct
"Todtenkopf" (death's head)	Likely the "Totenkopfschwärmer" (death's-head hawkmoth).	Correct
"Große Sumpfschnepfe" (large swamp snipe)	Modern term is "Große Brachschnepfe" (great snipe).	Correct

can make historical texts more interoperable and reusable, increasing the findability of specific species names and thereby making more knowledge about them accessible. Even with a certain margin of error, this approach remains a valuable tool to support human-in-the-loop data integration processes.

It is worth noting that the results presented here are not directly tied to a specific taxonomic resource. Nonetheless, aligning modern common names increases the likelihood of finding matches in existing taxonomies, facilitating better integration with contemporary databases. This research highlights a practical method for leveraging language models to automate the alignment of historical species names with standardized taxonomies.

Given the inherent risks of biases and errors in large language models, their usage bear risks. This study advocates for a human-in-the-loop approach, where human expertise complements the model's capabilities to ensure an effective mapping of historical terms to modern scientific names while mitigating risks associated with model inaccuracies.

6 Conclusion and Future Work

This research evaluated the potential of a large language model (LLM) to identify historical species names in biodiversity datasets using historical texts and expert annotations. Despite some challenges, the results revealed that 91% of common names and 96% of scientific names could be identified without context, with accuracy improving to 92% and 98%, respectively, when context was provided.

In addressing historical species naming challenges, the study identified five key issues: changes in spelling, the creation of new terms, shifts from broad historical common names to more specific modern equivalents (and vice versa), and the renaming of historical common names. The model successfully contributed to solutions for all five challenges, offering direct resolutions for four of them. These results highlight the significant potential of general-purpose LLMs to assist in making historical biodiversity data more findable, accessible, interoperable, and reusable (FAIR).

Future research may build on these findings by exploring several key areas. Comparative studies with other LLMs, particularly open-source models, could provide insight into their relative strengths and weaknesses for biodiversity data. Expanding the analysis to larger and more diverse datasets could better assess the scalability of the approach. Additionally, incorporating advanced techniques such as fine-tuning or retrieval-augmented generation could further enhance the model's performance, particularly in addressing complex challenges like aligning broad historical concepts with specific modern equivalents.

By continuing to refine and extend these methods, future research can better leverage LLMs to bridge historical and modern biodiversity data, ultimately contributing to more effective data integration and improved conservation research.

Literaturverzeichnis

[AL23] Ashok, Dhananjay; Lipton, Zachary C.: PromptNER: Prompting For Named Entity Recognition, 2023.

[Ba28] Barkham, Patrick: Country Diary 100 years on: sheep and dogs dominate over rabbits and house martins. The Guardian, S. 49–50, 2024-09-28.

[EF23] Elliott, Michael; Fortes, José: Using ChatGPT with Confidence for Biodiversity-Related Information Tasks. Biodiversity Information Science and Standards, 7:e112926, 2023.

[Ek23] Ekin, Sabit: Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices, 05 2023.

[Go24] Govaerts, Sander: Biodiversity in the Late Middle Ages: Wild Birds in the Fourteenth-Century County of Holland. Environment and History, 30(2):241–266, 2024.

[Ka24] Karam, Naouel; Fillies, Jan; Jonquet, Clément; Bouazzouni, Syphax; Löffler, Felicitas; Zander, Franziska; König-Ries, Birgitta; Güntsch, Anton; Diepenbroek, Michael; Paschke, Adrian: BiodivPortal: Enabling Semantic Services for Biodiversity within the German National Research Data Infrastructure. Datenbank-Spektrum, 24(2):129–137, 2024.

[Ko21] Kohlbecker, Andreas; Karam, Naouel; Paschke, Adrian; Güntsch, Anton: Preserving Taxonomic Change and Subsequent Taxon Relationships over Time. In: JOWO. 2021.

[Mi24] Minaee, Shervin; Mikolov, Tomas; Nikzad, Narjes; Chenaghlu, Meysam; Socher, Richard; Amatriain, Xavier; Gao, Jianfeng: Large Language Models: A Survey, 2024.

[Op24] OpenAI; Achiam, Josh; Adler, Steven; Agarwal, Sandhini; Ahmad, Lama; Akkaya, Ilge; Aleman, Florencia Leoni; Almeida, Diogo; Altenschmidt, Janko; Altman, Sam; Anadkat, Shyamal; Avila, Red; Babuschkin, Igor; more: GPT-4 Technical Report, 2024.

[Os23] Osawa, T; Tsutsumida, N et al.: The role of large language models in ecology and biodiversity conservation: Opportunities and Challenges. 2023.

[RG14] Rouhan, Germinal; Gaudeul, Myriam: Plant taxonomy: a historical perspective, current challenges, and perspectives. Molecular plant taxonomy: Methods and protocols, S. 1–37, 2014.

[Sv00] Svenonius, Elaine: The intellectual foundation of information organization, 2000.

[Vi22] Viana, Duarte S.; Blanco-Garrido, Francisco; Delibes, Miguel; Clavero, Miguel: A 16th-century biodiversity and crop inventory. Ecology, 103(10):e3783, 2022.

A Identification of Historic German Scientific Species Names

Table 8 displays subsections of the results when identifying historic scientific names.

Tab. 8: Extract of LLM based Identification of Historic German Scientific Species Names

Provided Text	LLM Prediction (Context)	LLM Prediction (No Context)	Annotated Scientific Names
An Fischen ist der Bezirk ebenfalls arm. In der Glatt findet sich die Bachforelle (Salmo Fario) und der Schuppfisch (Ciprinus Dobula), im Neckar außer letzterem auch die Barbe (Cyprinus Barbus), die Nase, gewöhnlich Weißfisch genannt (C. Nasus), die Groppe (Cottus Gobio) und der Aal (Muraena Anguilla).	Salmo Fario, Ciprinus Dobula, Cyprinus Barbus, C. Nasus, Cottus Gobio, Muraena Anguilla, Astacus saxatilis, Gommarus pulex	Salmo Fario, Ciprinus Dobula, Cyprinus Barbus, C. Nasus, Cottus Gobio, Muraena Anguilla, Astacus saxatilis, Gommarus pulex	Salmo Fario, Ciprinus Dobula, Cyprinus Barbus, C. Nasus, Cottus Gobio, Muraena Anguila, Astacus saxatilis, Gommarus pulex
Der große Spanner Gnophos furvata S. V. sitzt bei Tage an den Felsen und kann des Nachts mit der Laterne an Blüthen, besonders Echium, gesammelt werden. Von Kleinschmetterlingen werden aufgeführt: Grapholitha foenella L. s. tibialana Hb. 40, deren Raupe in den Wurzeln des Beifußes lebt, Grapholitha compositella Fab. als Gundiana Hb. 42, beide von Tuttlingen, und auch die seltene Myclois rosella Scop. pudorella Hb. 63.	Gnophos furvata, Grapholitha foenella, Grapholitha compositella, Myclois rosella	Gnophos furvata, Grapholitha foenella, Grapholitha compositella, Myclois rosella	Gnophos furvata S. V., Grapholitha foenella L. s. tibialana, Grapholitha compositella Fab., Myclois rosella Scop. pudorella
– Molytes dirus Hbst. auf dem Schloßberg; Grypedius equiseti Fabr. nicht selten an unserm größten Schachtelhalm auf einer feuchten Wiese unter Lustnau. – Daß Arten von Bostrychus in hiesiger Gegend durch ihre Menge besonderen Schaden den Waldbäumen zugefügt hätten, ist nicht bekannt geworden; wohl aber haben wir selber Äpfelbäume durch Eccoptogaster pruni Rtz. kranken sehen. – Von Bockkäfern mag erwähnt sein: Mesosa nebulosa Fabr.	Molytes dirus, Grypedius equiseti, Bostrychus, Eccoptogaster pruni, Mesosa nebulosa	Molytes dirus, Grypedius equiseti, Bostrychus, Eccoptogaster pruni, Mesosa nebulosa	Molytes dirus Hbst., Grypedius equiseti Fabr., Bostrychus, Eccoptogaster pruni Rtz., Mesosa nebulosa Fabr.

Investigating Zero-shot Topic Labeling of Scientific Papers Using LLMs

Jens Bruchertseifer [1], Patrick Neises [2], Maria Hinzmann [1], Ralf Schenkel [1], and Christof Schöch [1]

Abstract: In this paper, we focus on the problem of adding content labels of a given vocabulary to scientific publications using LLMs. After a short overview of the current state of the work, we present a first implementation of a zero-shot classification pipeline. This implementation is already realized with a focus on extendibility and customizability, so that it can easily be used for different data sets and use cases in the future. We select a subset of the DBLP Discovery Dataset and execute our pipeline on it. In the end, we discuss the results, suggest a comparison with a second data set, the STTCL journal from the humanities, and present its challenges. Both of the mentioned data sets comply with the FAIR data principles. Finally, we consider our plans for the next steps.

Keywords: LLM, automatic annotation, natural language processing, text analytics, annotation with a controlled vocabulary, AI-enabled methods, AI-readiness for research data, hierarchical multi-label classification, text classification, FAIR data principles

1 Introduction

Nowadays, the number of released scientific publications increases steadily. This development roots in different reasons, for example the rapidly growing number of researchers in most parts of the world. On the one hand, this advance is highly valuable. On the other hand, it makes it increasingly difficult to gather an overview of the research that has already been conducted. As a consequence, more and more work time is spent with creating such an overview instead of gathering new results. Therefore, the urgent need for a simplification of this process is very obvious. One such simplification would be the annotation of documents with topics from a controlled vocabulary. This would simplify the search for publications covering a specific topic. A controlled vocabulary offers better comparability and quality, since the opposite approach does not involve manual reviews.

The problem of automatically annotating documents with a controlled vocabulary is not new but has already been studied with multiple approaches (see for example [Fe97], [PSI06]). We try a modern approach, however, and discover a surprising result.

1 Trier University, Trier (Germany), jens.bruchertseifer@uni-trier.de, https://orcid.org/0000-0003-3536-3082;
hinzmannm@uni-trier.de, https://orcid.org/0000-0001-7199-1436;
schenkel@uni-trier.de, https://orcid.org/0000-0001-5379-5191;
schoech@uni-trier.de, https://orcid.org/0000-0002-4557-2753
2 Schloss Dagstuhl LZI, dblp group, Wadern (Germany),
patrick.neises@dagstuhl.de, https://orcid.org/0000-0002-3419-2544

doi:10.18420/BTW2025-122

Recent progress in the field of natural language processing provided us with a new technology known as *large language models (LLM)*. These models learn statistical relationships in their training corpus and are usually trained on large amounts of text data. A key feature of these models is that they can parse free-text prompts and even give responses to those. Furthermore, there can be contextual information provided in the query itself. For instance one could ask the model to summarize a paragraph of text that is provided in the query. It is important to note that these models are not simply trained on summarizing text but rather have a more universal set of skills.

In this paper we discuss how we can leverage a LLM in order to automatically annotate documents with topics from a controlled vocabulary. Therefore, we want to classify scientific publications from the computer science domain into topics from a controlled vocabulary. First, we derive a dataset of already labeled documents from the *DBLP Discovery Dataset (D3)*[3] created by Wahle et al. [Wa22], which provides automatically generated labels from the *Computer Science Ontology (CSO)*[4], a project created by Salatino et al. [Sa18]. We restrict our attention to only a small amount of the labels from the CSO, which form our controlled vocabulary of topics. We create a classification task and run two separate experiments of it, one using the titles as document representation, the other one using the abstract. Our workflow is to construct a set of prompts to provide the model with the possible topics and then ask the LLM which of those fits the given document best. In our experiments, we used the Llama 3 model, a state-of-the-art LLM published by Meta Platforms. Finally, we run an evaluation comparing the obtained annotation from the LLM approach with the already provided labels in the D3 dataset.

Our results show that the rate of publications that are annotated with a correct topic (according to our evaluation data) is the same for both titles and abstracts as document representation. Surprisingly, we observed that we obtain a significantly higher hallucination rate when abstracts are used instead of titles, which means that the LLM returns something that can not be matched to a topic from our controlled vocabulary. Both of these results were quite counter-intuitive to us because we initially assumed that the abstracts would yield richer semantic information than the titles and thus produce better results.

The paper is structured as follows. In Section 2, we discuss the current state of research regarding automatic annotation with controlled vocabularies. Section 3 discusses how we obtained the dataset used in this evaluation. In Section 4, we briefly describe how we constructed the prompts for the LLM and mapped the results on the controlled vocabulary. The technical details of our implementation and the LLM used in this project can be found in Section 5. An evaluation of the results obtained by the annotation with the LLM is discussed in Section 6. Finally, Section 7 concludes the paper and discusses potential future work.

3 https://zenodo.org/records/7071698
4 https://cso.kmi.open.ac.uk/about

2 Related work

There has been a long interest in adding topic labels to texts, also known as text classification or topic labeling. Classic approaches for topic labeling used, for example, support vector machines (SVM) or naive Bayes classifiers. For a more detailed overview, we refer to e. g. [Ko19]. However, since the recent rise of Deep Learning, the focus has shifted to such approaches, e. g. transformers like BERT [De19] or large language models (LLMs), cf. [Li22], [Ta24b] or [ZJN21].

Obviously, a "non-hierarchical single-label classification" approach such as the one presented below can also be seen as a special case of *hierarchical multi-label classification* (where in general the labels represent the leaf nodes of a multi-tier hierarchy tree and documents are always labeled with the leaf nodes and also the nodes of the path): The used labels from the CSO are nothing less than one layer of the label hierarchy of the CSO and since there is only one such layer within our labels, a path would simply contain not more than one node: A leaf node. In [SC22], Sadat et al. used the ACM CCS tree as labels when trying a similar idea exclusively on computer science literature. Hierarchical multi-label classification has been implemented even at an industrial scale, see [Ta24a]. Martorana et al. tried a variant by adding labels from a controlled Linked Data vocabulary to column headers using LLMs [Ma24]. For our early approach, a single layer of labels will be sufficient, but hierarchical levels of labels will be a relevant option for future work.

It should be noted that the term "topic" is used for multiple concepts: The well-known discipline of topic modeling uses it for sets of terms which are automatically generated during the execution. Our method however uses existing and established vocabularies, which has the advantage that we do not introduce any new incompatibilities with the existing vocabulary. It can be noted that our use case can also be distinguished from Named Entity Recognition (NER), since the topics might not be mentioned explicitly in the title or abstract.

3 The dataset

In order to test an automatic annotation approach based on LLMs, we needed some data that is already labeled in some form. For obtaining such a dataset, we used the aforementioned dataset D3. It contains all papers and authors extracted from the *dblp computer science bibliography*,[5] which is operated by NFDI4DS and NFDIxCS member Schloss Dagstuhl LZI,[6] and enriched these records with further metadata: For example, abstracts have been added to publication records. Since its original release, there have been two new versions of the dataset, where the latest version (2.1) incorporates labels from the aforementioned CSO. More information about the new versions of D3 can be found on the project GitHub page.[7]

5 https://dblp.org

6 https://blog.dblp.org/2021/07/02/schloss-dagstuhl-becomes-part-of-the-national-research-data-infrastructure-for-data-science-and-artificial-intelligence/

7 https://github.com/jpwahle/lrec22-d3-dataset

The CSO is a large scale hierarchical ontology of research areas in computer science, but also in other domains as well: Therefore, the ontology contains secondary root nodes like "linguistics" or "geometry" in addition to the self-explanatory root node "computer science". At the time of writing, "computer science" has 19 subordinate topics, which are:

- artificial intelligence
- computer aided design
- computer hardware
- computer imaging and vision
- computer networks
- computer programming
- computer security
- computer systems
- data mining
- human computer interaction
- information retrieval
- information technology
- internet
- operating systems
- robotics
- software
- software engineering
- theoretical computer science
- bioinformatics

These 19 topics form the controlled vocabulary for the annotation process in our approach.

The documents in the D3 dataset have been labeled with topics using the CSO classifier.[8] This classifier has two separate modules, the syntactic and the semantic one, and outputs four sets of labels. The first two sets correspond to the outputs of each module, the third is their union and the fourth includes the relevant super-areas in the ontology. Thus, when building the union over all four lists, we should have at least one of the high level topics presented above for every classified document.

By using this assumption, we created our test dataset from the D3 dataset in the following way: We considered all publications in the publications part of the dataset. For every publication, we computed the union of all four resulting lists of the CSO classification. Then, we discarded all elements of this list besides the 19 high level topics we discussed earlier. If the list obtained this way has no elements, the publication is discarded. The minimal number of topics is thus one. It should be noted that a publication can have more than one topic associated with itself: The maximal number of topics observed in our dataset is twelve. On average, a publication has approximately 4.5 topics. Then, we filtered out all publications that did not have a title or an abstract. From the resulting publications, we selected 2500 at random, which were used to form our dataset.

The extracted dataset is saved in a JSON file. This file contains a list of dictionaries, where each dictionary represents one of the 2500 randomly selected publications with a small set of metadata, which can be seen below. For our purposes, we saved the `corpusid`, which is the identifier of the publication in the D3 dataset under the `D3 ID` key and saved the title,

8 `https://github.com/angelosalatino/cso-classifier`

```
{
  "D3 ID": 18298,
  "title": "Harnessing the power of two crossmatches",
  "abstract": "Kidney exchanges allow incompatible donor-patient ..."
  "subjects": [
    "computer hardware",
    "computer networks",
    "computer systems",
    "information retrieval",
    "information technology",
    "internet",
    "theoretical computer science"
  ]
}
```


Fig. 1: First publication record of the dataset extracted from D3 (abstract shortened)

the abstract and the extracted topics in the `title`, `abstract` and `subjects` keys respectively. An example of such a publication entry is given in Figure 1. Even though the abstract might seem like a misclassified item from medicine at first glance, it is indeed a computer science publication.

4 Methodology

Now that we have an understanding of how our data was obtained, we will discuss the methodology for the annotation process using an LLM. In order to annotate our documents, we need to create a set of prompts for each document that is processed by the LLM, such that we obtain a "parsable" answer from which we can extract the annotated data. For the setting presented in this paper, we focused on using the titles and abstracts in separate experiments as document representation data and wanted to obtain one of the topics from the controlled vocabulary constructed in Section 3 that fits to the document in each experiment. We decided to use three prompts for every document: The first two are used to provide the model with our controlled vocabulary, the 19 subordinate topics of "computer science" from the CSO, and the last one asks to annotate the current document. The concrete prompts (in the version for the experiment using the title as document representation) are presented in Figure 2, where `{document representation}` represents only a placeholder for this figure and is substituted with the title of the current document in the implementation. The prompts for the experiment based on the abstracts are analog in structure: We only replaced the word "title" with the word "abstract" in the static part of the prompt and provided the LLM with the abstract of the document instead of the title at the `{document representation}` placeholder.

These three prompts were evaluated for every document in our dataset. As it can be seen in the last query, we informed the LLM that we only wanted one topic from our controlled

We want to create a list of topics in the following. We call this list targets_list.

Here are some topics that should be added to the targets_list: artificial intelligence, computer aided design, computer hardware, computer imaging and vision, computer networks, computer programming, computer security, computer systems, data mining, human computer interaction, information retrieval, information technology, internet, operating systems, robotics, software, software engineering, theoretical computer science, bioinformatics. Please use the exact spelling that I provide to you.

We now want to annotate a title with the topics provided in the targets_list.
Given the following title: `{document representation}`
Please assign 1 suitable topic from the targets_list to the title.
This topic should be contained in the targets_list we created earlier and use the exact spelling of the topic in the targets_list.
Please respond only with the 1 topic without any further text.

Fig. 2: Prompts used for this work where the term `{document representation}` represents a placeholder which is replaced with the actual title in every request

vocabulary as an answer. Then, we took the result of the third query and tried to match it with our controlled vocabulary in the following way: We split the result string at the symbol "," and treated each of the elements obtained that way as a candidate string that should be matched to our controlled vocabulary. Then, we transformed each of these candidate strings to lower case and removed all whitespace at the beginning and the end. It should be noted that the string splitting and the removal of the whitespaces was done because the code used for this evaluation is also capable of asking the LLM for more than one topic and in that case, it expects a comma-separated list as result, whereas the normalization in terms of converting the string to lower case was introduced because we observed that some models used capital letters in the response. We matched a candidate string to an element in the controlled vocabulary if the strings were equal.

5 Implementation

The code together with the instructions on how to perform our analysis is provided on GitHub.[9] To foster its further usage, we have put efforts in making the tool *FAIR* (findable, accessible, interoperable and reusable) [Wi16] by using an open license, hosting the code on GitHub and also the usage of open tools and packages. The code is written in Python, using the GPT4All package[10] [An23] for all LLM related aspects.

9 https://github.com/paNeises/paper-2025-bigds
10 https://pypi.org/project/gpt4all/

For better reproducibility, self-hosted solutions were preferred instead of cloud-based services like ChatGPT or Claude. From these self-hosted software suites, we preferred a tool that not only could provide access to multiple LLMs but also could be accessed via existing SDKs, bindings or program libraries, and was likewise mature and comparably well-documented. Taking these requirements into account, GPT4All fulfilled our requirements. We used it via its Python bindings (version 2.8.2).[11]

In order to annotate the dataset, our implementation constructed the three prompts described in Section 4 for every document in the dataset. The prompts were then evaluated with an LLM using a separate chat session for every document. GPT4All allows to use multiple LLMs out of the box. It has to be noted that there exists the option to include further user-embedded LLMs, which is another interesting question for future work. Therefore, it was not possible to test every LLM that could run with GPT4All. We performed tests with predefined LLMs and eventually chose Llama for this first implementation, since it showed decent results. We used the Meta Llama 3 model (`Meta-Llama-3-8B-Instruct.Q4_0.gguf`), which is already preconfigured as one option with the GPT4All Python bindings.

For each document, the annotated keywords obtained by the LLM were then added to the dictionary representing its metadata as a new field and the final list containing all documents was stored. Finally, we ran an evaluation on the previously saved annotated set of documents, whose results are covered in Section 6.

6 Evaluation

The annotation as well as the evaluation were executed on a workstation equipped with an AMD Ryzen 9 7900X with twelve cores and 24 threads, 128 GB of RAM, a single Nvidia RTX 4090 and a 2 TB M.2 PCIe SSD.

In order to evaluate the annotated data obtained by the LLM, we differentiated the following three cases when observing the annotation result:

- A topic from our provided topic list was successfully extracted from the LLM response and this topic is in the subjects-list in the document metadata. We call this case *success*.
- A topic from our provided topic list was successfully extracted from the LLM response but this topic is not in the subjects-list in the document metadata. We call this case *misclassified*.
- It was not possible to extract a topic that matches one of the entries in our topic list from the LLM response. We call this case *hallucination*, which might differ a bit from its common usage.

11 Please beware that the semantic versioning of GPT4All differs from the semantic versioning of the Python bindings published on PyPi. We refer to the version of the Python bindings, cf. `https://pypi.org/project/gpt4all/#history`, which have not yet reached version 3 and beyond.

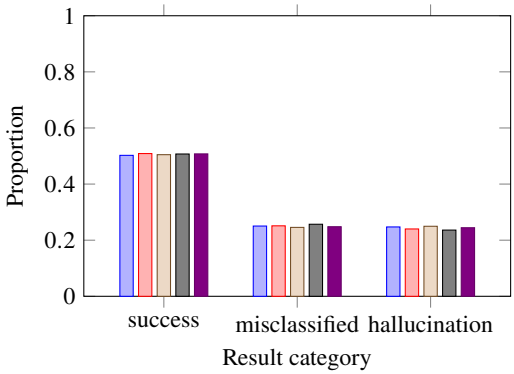


Fig. 3: Analysis of the LLM annotation on the dataset extracted from D3 based on titles (multiple runs indicated by color; each bar represents the proportion of a result category from a single run)

It should be noted that after the matching step, the returned list of keywords that were annotated by the LLM contains one element for the first two cases, whereas the last case always implies that the list is empty. As previously mentioned, we conducted two separate experiments, the first of which using the document title as the document representation and the second one using its abstract. The workflow of these experiments was already discussed in Section 4 and Section 5. We gathered the results from the annotation process using the LLM and compared the results with the topics in the subject-lists in the document metadata extracted from D3 in the process of creating the dataset as discussed in Section 3. In every experiment we performed five separate runs in order to see if there is a variation between the results of these runs.

The results of the first experiment (based on titles) are shown in Figure 3. As we can see, the results remained fairly consistent between the runs in terms of numbers. We observed the success case in around 50 % of the documents labeled by the LLM across all runs. The misclassified case, on the other, hand occurs in 25 % of the documents. For 25 % of the documents, the annotation by the LLM ends up in the hallucination case. The running time of the annotation process of all 2500 documents also remained consistent across all runs and took us around 2 hours and 52 minutes for each run.

The results of the second experiment based on abstracts are shown in Figure 4. Similarly to the results based on the titles, we also observe that the results stayed consistent between the different runs in terms of numbers. Also, the rate of the success case did not change and stayed at around 50 %. On the other hand, we did not observe that many misclassified cases in the second experiment, as their rate is only about 17 %. This leaves around 33 % of the documents, for which we ended in the hallucination case. The running time again stayed consistent between all runs and took us about 3 hours and 30 minutes for each run.

These results were quite surprising to us. We did not anticipate the following three aspects:

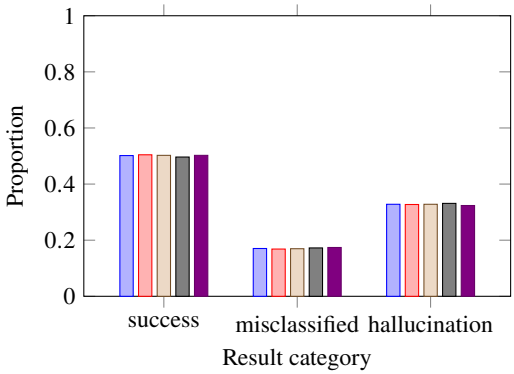


Fig. 4: Analysis of the LLM annotation on the dataset extracted from D3 based on abstracts (multiple runs indicated by color; each bar represents the proportion of a result category from a single run)

On the one hand, we did not expect the decrease of the success rate, since the abstracts usually provide much more details and context than the title, which could help to classify the documents correctly. On the other hand, the increase in the hallucination rate itself was unexpected. Our assumption was that it would at least stay constant or even go down, since the requirement, that the given vocabulary had to be used, did not change. At last, we did not assume that the number of results would shift so significantly from results of the case misclassified to results of the case hallucination, as we anticipated a growth of the success rate.

We further inspected some of the responses that led to a hallucination case by hand using a separate annotation run based on titles. For example, we observed in one case that the LLM returns "'software engineering'" (using the inner quotes; outer quotation marks due to the quotation), which prevents our matching step from finding the correct topic since it does not remove quotes. Further, we observed multiple responses from the LLM that were "computer vision". However, this topic does not exist in our controlled vocabulary but "computer imaging and vision" does. In these cases, it would be possible to resolve these cases with a better matching step. Nevertheless, by far the most answers, that we observed and that ended up in the hallucination case, were "computer science".

From a statistical point of view, a random guessing algorithm would provide about 24 % success rate, since we have 19 topics and each document is annotated with 4.5 topics on average. Compared to such an algorithm, this rate is significantly better, even though it is not as high as we expected it to be. This leaves room for discussion and considerations to improve this experimental workflow. Possible reasons might be that the subject areas of scientific topics are too small and insufficiently covered in the training data sets, so that LLMs do not reach the results achieved in other areas.

7 Discussion and future work

As shown in Section 6, we provided a zero-shot annotation approach that significantly outperforms a random guessing algorithm on the discussed dataset. We present it as an early-stage approach together with the lessons learned and are looking for scientific feedback for further development. The results indicate room for improvement of the presented approach. In the following, we discuss some ideas for future work.

Our results show that the rate of publications annotated with a correct topic stays the same for both titles and abstracts as document representations. An explanation might be that more input also leads to more possibilities that could "confuse" the model, which is an interesting question for future research. Also, we observed two limiting factors for our approach: The *suggested maximal prompt length* (maximum length of a prompt that the LLM was trained for) and its *context length* (maximum previous input/output that the LLM considers for answers). However, the documentation lacks information about them. We only observed these limits in the debug messages, rises interesting questions for the future. Additionally, we only focused on a small set of topics extracted from a large scale ontology. In the future, we want to explore the scalability in terms of the number of topics.

Another interesting question is if the model can correct itself in the case of a hallucination. This case is easily detectable because we can apply our matching step and consider an empty result as hallucination, since we did not obtain a topic from our controlled vocabulary. The idea is for such cases to supply an additional prompt to the LLM that points to the erroneous output and asks the LLM to correct it to a topic from the provided vocabulary.

Our intention was to create this framework as customizable as (sensibly) possible, so that it could be used by other people in different domains as well. We have also gained the impression that LLMs for topic labeling are not frequently used in the humanities so far. Therefore, we wanted to test it also with data from the humanities, but it turned out that such (coherent) digital text or metadata sets together with suitable topic vocabularies are quite rarely available and are even more unlikely to have an open license. We found the journal "Studies in 20th & 21st Century Literature (STTCL)",[12] which uses an open access license. Its articles are already labeled with topics from the Digital Commons Network™ (DCN),[13] which seemed suitable for this approach. We took the union of all DCN topics annotated to STTCL documents as the set of which the LLM had to choose a valid topic. First tests with the titles of all items of the journal (the issues include book reviews, editor's notes, etc.) showed only a success rate of about 10 %, which might indicate that this data sets requires some preprocessing before it is usable. Therefore, we plan to tackle this approach again with a further processed version of the STTCL data set in the future. As a first result from these investigations, it can be concluded that (open) data sets in the humanities are often in an earlier stage than data sets in more computer-related disciplines and might require more manual preprocessing.

12 https://newprairiepress.org/sttcl/
13 https://network.bepress.com/

Bibliography

[An23] Anand, Yuvanesh; Nussbaum, Zach; Treat, Adam; Miller, Aaron; Guo, Richard; Schmidt, Ben; Community, GPT4All; Duderstadt, Brandon; Mulyar, Andriy: GPT4All: An Ecosystem of Open Source Compressed Language Models. CoRR, abs/2311.04931, 2023.

[De19] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In (Burstein, Jill; Doran, Christy; Solorio, Thamar, eds): Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, pp. 4171–4186, 2019.

[Fe97] Ferber, Reginald: Automated Indexing with Thesaurus Descriptors: A Co-occurence Based Approach to Multilingual Retrieval. In (Peters, Carol; Thanos, Costantino, eds): Research and Advanced Technology for Digital Libraries. First European Conference, ECDL '97, Pisa, Italy, 1-3 September, Proceedings. volume 1324 of Lecture Notes in Computer Science. Springer, pp. 233–252, 1997.

[Ko19] Kowsari, Kamran; Meimandi, Kiana Jafari; Heidarysafa, Mojtaba; Mendu, Sanjana; Barnes, Laura E.; Brown, Donald E.: Text Classification Algorithms: A Survey. Inf., 10(4):150, 2019.

[Li22] Li, Qian; Peng, Hao; Li, Jianxin; Xia, Congying; Yang, Renyu; Sun, Lichao; Yu, Philip S.; He, Lifang: A Survey on Text Classification: From Traditional to Deep Learning. ACM Trans. Intell. Syst. Technol., 13(2):31:1–31:41, 2022.

[Ma24] Martorana, Margherita; Kuhn, Tobias; Stork, Lise; van Ossenbruggen, Jacco: Zero-Shot Topic Classification of Column Headers: Leveraging LLMs for Metadata Enrichment. In (Salatino, Angelo A.; Alam, Mehwish; Ongenae, Femke; Vahdati, Sahar; Gentile, Anna Lisa; Pellegrini, Tassilo; Jiang, Shufan, 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. volume 60 of Studies on the Semantic Web. IOS Press, pp. 52–66, 2024.

[PSI06] Pouliquen, Bruno; Steinberger, Ralf; Ignat, Camelia: Automatic annotation of multilingual text collections with a conceptual thesaurus. CoRR, abs/cs/0609059, 2006.

[Sa18] Salatino, Angelo Antonio; Thanapalasingam, Thiviyan; Mannocci, Andrea; Osborne, Francesco; Motta, Enrico: The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas. In (Vrandecic, Denny; Bontcheva, Kalina; Suárez-Figueroa, Mari Carmen; Presutti, Valentina; Celino, Irene; Sabou, Marta; Kaffee, Lucie-Aimée; Simperl, Elena, eds): The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II. volume 11137 of Lecture Notes in Computer Science. Springer, pp. 187–205, 2018.

[SC22] Sadat, Mobashir; Caragea, Cornelia: Hierarchical Multi-Label Classification of Scientific Documents. In (Goldberg, Yoav; Kozareva, Zornitsa; Zhang, Yue, eds): Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, pp. 8923–8937, 2022.

[Ta24a] Tabatabaei, Seyed Amin; Fancher, Sarah; Parsons, Michael; Askari, Arian: Can Large Language Models Serve as Effective Classifiers for Hierarchical Multi-Label Classification of Scientific Documents at Industrial Scale?, 2024.

[Ta24b] Taha, Kamal; Yoo, Paul D.; Yeun, Chan Yeob; Homouz, Dirar; Taha, Aya: A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. Comput. Sci. Rev., 54:100664, 2024.

[Wa22] Wahle, Jan Philip; Ruas, Terry; Mohammad, Saif M.; Gipp, Bela: D3: A Massive Dataset of Scholarly Metadata for Analyzing the State of Computer Science Research. In (Calzolari, Nicoletta; Béchet, Frédéric; Blache, Philippe; Choukri, Khalid; Cieri, Christopher; Declerck, Thierry; Goggi, Sara; Isahara, Hitoshi; Maegaard, Bente; Mariani, Joseph; Mazo, Hélène; Odijk, Jan; Piperidis, Stelios, eds): Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022. European Language Resources Association, pp. 2642–2651, 2022.

[Wi16] Wilkinson, Mark D.; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Olavo Bonino; Bourne, Philip E.; Bouwman, Jildau; Brookes, Anthony J.; Clark, Tim; Crosas, Mercè; Dillo, Ingrid; Dumon, Olivier; Edmunds, Scott C.; Evelo, Chris T. A.; Finkers, Richard; González-Beltrán, Alejandra N.; Gray, Alasdair J. G.; Groth, Paul; Goble, Carole A.; Grethe, Jeffrey S.; Heringa, Jaap; 't Hoen, Peter A. C.; Hooft, Rob W. W.; Kuhn, Tobias; Kok, Ruben G.; Kok, Joost N.; Lusher, Scott J.; Martone, Maryann E.; Mons, Albert; Packer, Abel Laerte; Persson, Bengt; Rocca-Serra, Philippe; Roos, Marco; van Schaik, Rene C.; Sansone, Susanna-Assunta; Schultes, Erik Anthony; Sengstag, Thierry; Slater, Ted; Strawn, George O.; Swertz, Morris A.; Thompson, Mark; van der Lei, Johan; van Mulligen, Erik M.; Velterop, Jan; Waagmeester, Andra; Wittenburg, Peter; Wolstencroft, Katy; Zhao, Jun; Mons, Barend: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3, 2016.

[ZJN21] Zhang, Shikun; Jafari, Omid; Nagarkar, Parth: A Survey on Machine Learning Techniques for Auto Labeling of Video, Audio, and Text Data. CoRR, abs/2109.03784, 2021.

ADISS: Authority Data Integration Search System

Leon Fruth [1], Tobias Gradl [1], and Andreas Henrich [1]

Abstract: This paper introduces ADISS, a generic search system designed to integrate heterogeneous authority file providers. Authority data is used to unambiguously identify entities such as persons, places, and organizations. As individual data providers usually do not offer both broad, universal data and domain-specific, in-depth data at the same time, in many application scenarios a combined access to multiple datasets is required to support real-world use cases. In the context of Digital Humanities this combination of multiple authority data providers improves the resolution of ambiguities in data curation processes. Our work in this direction is mainly motivated by two projects that require semi-automatic retrieval, as well as user-centered search scenarios for different authority file providers. Instead of using multiple existing endpoints to access the various datasets, we gather the heterogeneous data and make it accessible via integrated query and result models, thus simplifying access at the level of technical interfaces and schema integration. In this paper, we present our highly configurable search API, which offers a diverse range of search and filtering options. We show that by its generic and highly configurable nature, our system is adaptable and reusable for a diverse set of use cases and conclude the paper with ideas for further steps and improvements.

Keywords: FAIR, Digital Humanities, Authority Data, Data Curation

1 Introduction

The FAIR acronym, which stands for Findability, Accessibility, Interoperability, and Reusability of digital assets, has become an important concept in academia, serving as a crucial criterion for effective research data management [Wi16]. Authority files play a key role regarding the implementation of the FAIR principles by enabling unambiguous identification of entities like locations, organizations or persons. Rather than relying solely on textual descriptions, links to entity representations in knowledge bases enrich and contextualize research data and thus enhance their overall interoperability, making it easier to discover and use [Ha21].

There are several sources for authority data, each containing distinct sets of entities and focusing on different entity attributions that are influenced by the legal and contextual framework of the respective provider. These providers can be broadly categorized into three groups. The first group consists of highly regulated authority file providers such as the Gemeinsame Normdatei (GND)[2] or the Virtual International Authority File (VIAF) that are managed by trusted institutions and libraries. The GND is a project of the German National

[1] University of Bamberg, Media Informatics Group, An der Weberei 5, 96047 Bamberg, Germany,
leon.fruth@uni-bamberg.de, https://orcid.org/0009-0001-2128-3025;
tobias.gradl@uni-bamberg.de, https://orcid.org/0000-0002-1392-2464;
andreas.henrich@uni-bamberg.de, https://orcid.org/0000-0002-5074-3254

[2] https://www.dnb.de/DE/Professionell/Standardisierung/GND/gnd_node.html All links accessed on 23-12-2024

doi:10.18420/BTW2025-123

Library (DNB) and is strictly maintained within quality standards and legal settings [BP11]. The VIAF is a collaborative effort by the libraries DNB, the Library of Congress (LOC) and the Online Computer Library Center (OCLC) to create an overarching authority file which spans multiple national libraries and archives [Be06]. Due to their focus on quality and institutionally induced biases[3], providers like the GND expose sets of highly regulated and curated entries that are naturally limited in quantity.

The second group includes community-based projects such as Wikidata[4] which hold substantial amounts of user-generated data. Wikidata is commonly utilized in Digital Humanities (DH) projects to support metadata curation, annotation and Named Entity Recognition (NER) [Zh22]. Additionally, Wikidata is also utilized in natural sciences to improve data integration and accessibility by identifying people that are associated with entities, such as specimens and taxonomic names [Gr20; Gü21]. Despite unmatched degrees of quantity, these community-based platforms pose challenges related to data accuracy and exhibit biases driven by public interests. [Zh22]

Lastly, the third group consists of small, specialized data repositories. For instance, Memorial Archives[5] contains entity descriptions specific to the Flossenbürg concentration camp, while the historic place names in Bavaria from Geschichte Bayerns[6] can be a valuable resource for certain research communities. Often being very detailed in their respective contextual setting, specialized data repositories attempt to fill specific gaps, but do not contain descriptions of more general entities.

As a result of these limitations, relying on a single authority data provider often does not satisfy the requirements of DH projects. Integrated access to multiple data providers on the other hand improves curation possibilities – enhancing the quality of the structure, detail, and semantic richness of the data and thus its findability [HG21]. Such requirements on integrated access to several authority file providers for two DH projects will be elaborated in the next section.

Thereafter, this paper introduces the Authority Data Integration Search System (ADISS), which integrates various heterogeneous authority file providers and provides a wide range of search and retrieval techniques. Its search component is based on preliminary work in the infrastructural contexts of DARIAH-DE and CLARIAH-DE [GH16].

[3] National and cultural biases in the case of national libraries due to their mission
[4] https://www.wikidata.org/
[5] https://memorial-archives.international/
[6] https://www.geschichte-bayerns.de/ortsnamen

2 Motivation

The idea of consolidating access to multiple authority file providers results from the informational needs in the research infrastructures Text+[7] and Oral-History.Digital (oh.d)[8]. Despite differences in contexts, user communities and usage scenarios, both initiatives share the goal of enhancing the visibility and findability of research data, and have similar requirements related to the retrieval and use of authority data.

The project oh.d develops and operates a data curation and research platform for collections of audio-visually recorded narrative interviews. The metadata of these interviews typically include attributions in relation to entities, such as the names of interviewer and interviewee and other referenced persons, organizations and locations, often historical places where the interview was conducted or to which the interviewee is connected. Transcriptions are available for a majority of the interviews in oh.d and include large amounts of NER and manually extracted named entities that should be linked to authority data. Interviews in the context of oh.d are mostly life history interviews, which – while often focusing on the individual's entire life – are linked to a specific thematic context, such as the Holocaust in Germany or life as a migrant worker. Access to multiple data providers facilitates referencing of both generic and very specific entities which are typically not found within the dataset of one single provider. Furthermore, some of these data providers do not offer features such as consistent coordinates or labels in multiple languages, which are functionally required in the oh.d portal. Interview data is imported from existing databases and manually edited in the oh.d portal, which requires support for authority data search scenarios in both semi-automated and manual settings.

Text+ is a consortium in the German National Research Data Infrastructure (NFDI). It focuses on language and text resources that are of high relevance in related disciplines, and aligns them along the categories of digital collections, lexical resources and editions. A central component of the Text+ infrastructure is the Text+ Registry[9], which serves as a central resource catalog. It builds on resource descriptions scattered across existing catalogs and data sources and facilitates the manual addition and enrichment of metadata. A fundamental aspect of enrichment is the contextualization of resources, i.e. the explication of references between resources (e.g. the edition of a particular letter being part of the complete edition of all works of an author), and to related entities such as persons, organizations and locations. Much like the oh.d portal, the Text+ Registry requires support for semi-automatic correlation of entities when importing metadata from existing catalogs. First the entities are imported automatically using the metadata, which can be names of entities, or identifiers from a authority file, then the resolved entities need to be manually confirmed for correctness. Furthermore, imported entries and resources missing in connected catalogs are manually curated by domain experts and require user-centered search facilities for suitable authority data.

[7] https://text-plus.org/
[8] https://www.oral-history.digital/
[9] http://registry.text-plus.org

Both Text+ and oh.d implement functionalities to contextualize data by means of identifying and explicating relations to various classes of entities. Focusing academic contexts and users, both infrastructures prefer information of high-quality and domain-specific authority file providers, but require their combination with community-based sources to be able to search in a large set of entities. As this combination requires processing of multiple query and response formats and the identification of duplicates between sources, an implementation in individual project contexts seems redundant and impractical. With requirements for both semi-automatic and user-centered authority data search scenarios, the presented use cases are examples for projects with similar requirements, which allows ADISS to be designed to support the specific needs of these projects and to be generic and reusable.

3 Data Providers

The selection of the initial set of authority file providers for the implementation of ADISS has been influenced by the needs of the above usage scenarios. To determine the general feasibility, scalability and robustness of the proposed solution, we have initially focused on the distinct entity type of locations and gathered data from multiple providers – commencing with the relevant and large datasets of GND, Wikidata, OpenStreetMap (OSM) and Geonames.

The central authority file provider used in the previously described use cases is the GND provided by the DNB. The GND consists of six different entity types and close to 10 million total entities (as of 18.12.2024). The data can be obtained as a data dump[10] or queried via a web-API[11]. The API allows entities to be retrieved by their identifier and through a search function, which includes basic filtering and sorting options. However, it lacks features such as fuzziness, suggestions and geographical filters. Additionally, the data does neither fully support multilingual names and descriptions nor does it contain sufficient geographical information. However, other authority data sources can be used to fill these gaps for entities that exist in multiple sources and contain at least an unidirectional reference to the respective GND entry.

A rather large data source is Wikidata with over 100 million total entities. Within its extensive collection, Wikidata offers a high quantity of labels, aliases and descriptions in numerous languages along with annotated language codes. This comprehensive dataset provides access to a wealth of information that cannot be found on a comparable scale in other sources such as the GND or similar providers. Moreover, Wikidata also features a large amount of reference links and identifiers connecting it with various other websites and authority data providers, including VIAF, GND, OSM and Geonames. These connections enable the linking of entities from different sources which enhances overall knowledge integration. Due to their community-based nature, Wikidata and other providers have some

[10] https://data.dnb.de/opendata/
[11] https://lobid.org/gnd/api

quality issues, such as (near) duplicate entries or incorrect links to other authority data providers [Me21]. The data can be accessed through dumps in multiple formats[12] as well as via SPARQL-endpoint[13] or the Wikibase REST-API[14]. Still, these endpoints do not support options like fuzziness or geographical filtering at present.

An additional data provider utilized for the described use cases is OSM[15], which includes detailed coordinates of locations and regions but overall with varying quality of data. OSM data can be queried through different APIs: a prominent example is Nominatim[16], which offers text searches, but lacks functionalities such as fuzziness and geographical filtering. Another option can be found in Photon[17], a service that allows some geographical prioritizing, but does not include many language labels that are present in Nominatim.

Geonames[18] is another geographical database that contains over 11 million places. The data is accessible through a web API that offers full-text search capabilities; however, it misses certain features like the aforementioned APIs. Both OSM and Geonames offer complete datasets as downloadable dumps for further usage.

4 Background and Architectural Considerations

Building a system that allows querying the above mentioned datasets in an integrated manner can be approached in different ways. One option is implementing a federated meta-search that queries and aggregates over multiple of these endpoints. However, this approach is limited by the capabilities of the underlying APIs, lacking in required search functionalities necessary for the previously described use cases. Moreover, ranking results from different APIs is challenging due to differences in data types, query languages and retrieval models between providers.

The second approach, which is presented in this work, involves gathering the data from multiple data sources and integrating it into a homogeneous data model. By indexing the data, we can offer extensive search functionality through a single API while also accessing various data sources concurrently. This method provides arguably better performance than the previously mentioned approach as it reduces network latency by minimizing requests down to only one instead of N requests for each of the N considered data providers.

Moreover, heterogeneous data is integrated and cached into a homogeneous format during indexing rather than during runtime when the providers are queried, which further improves

[12] https://dumps.wikimedia.org/wikidatawiki/entities/
[13] https://query.wikidata.org/
[14] https://www.mediawiki.org/wiki/Wikibase/API
[15] https://www.openstreetmap.org
[16] https://nominatim.org/
[17] https://photon.komoot.io/
[18] https://www.geonames.org/

the performance. Additionally, highly specialized data sources that lack an API can be added for research projects requiring such information.

For scalability and reduced maintenance efforts, a hybrid approach combining download and API-access could also be used, which has been presented in [Je23]. Here an upstream search API has been created for some providers while existing endpoints have been used for others. Similarly, the GFBio Terminology Service integrates various terminologies either internally indexed or externally accessed [KLM17]. In general, for some providers without a sufficiently powerful search interface, a new central index can be created on the basis of the downloads provided, combined with a meta-search for other providers with large data sets and sufficiently powerful search functionality. Such an approach could also be considered for ADISS in the future if scalability and up-to-dateness pose a problem, or if a data provider does not offer its data for download but allows API-based search access. However, there are currently no problems in this respect. The data is regularly updated to meet the needs of the above usage scenarios, and the combined storage requirement for the five integrated databases is 67 gigabytes.

The following section outlines the implementation for our proposed solution ADISS, not only addressing the requirements from oh.d and Text+, but also presenting a generic solution expandable to other fields and use cases.

5 Implementation

This section presents central aspects of the implementation of ADISS. Figure 1 shows the architecture of the service. For each data provider, an individual data wrapper processes the respective format and structure. The first three steps in the data pipeline are considered in the first subsection 5.1. In the subsequent subsection 5.2, the concept for the integrated data model is outlined. In subsection 5.3 the Generic Search (GS) API is presented with its most relevant features and configuration options. This search component provides search and retrieval functionalities for the processed data.

5.1 Data Collection and Processing

For each data provider that is required in the application context and offers download options for their authority data, an individual data wrapper[19] is implemented using Java and the Spring Boot Framework. These wrappers share core functionalities, but are adapted to the respective structural constraints of the data. Additionally, existing functionality in libraries like osm2pgsql[20] for OSM data dumps or the Wikidata Toolkit[21] for Wikidata have been reused for data processing.

[19] https://gitlab.com/minfba/resinfra/adiss
[20] https://osm2pgsql.org/
[21] https://www.mediawiki.org/wiki/Wikidata_Toolkit

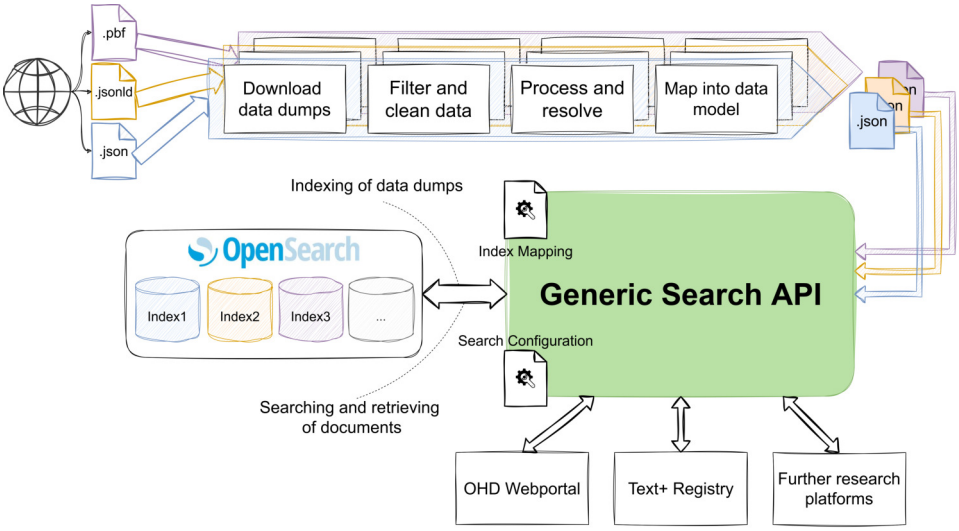


Fig. 1: The integrated authority data service.

The process starts with downloading the data dump from its respective source in the available format, GND entities in JSON-LD format, OSM data dumps in PBF format, and Wikidata in JSON or XML dumps. Next, the extracted data is filtered to include only entries of relevance for our use cases. In Wikidata, for instance, entities that are subclasses of irrelevant entries such as astronomical objects[22] or chemical compounds[23] are not considered. In addition, irrelevant attributions and media contents, are removed. With OSM, entries like highways or power lines are ignored.

Moreover, some fields are required to be structurally adapted; for example, coordinates which are represented in varying standards such as well-known text (WKT), are all transformed into the GeoJSON format. To facilitate comprehensive search capabilities on data from multiple sources, similar harmonization needs to be applied to other data fields like the entities *type* or *classes* of individual data providers and country- or subdivision-codes, which are also present in different ISO-formats.

5.2 Data Model

After data downloading and preprocessing, a subsequent task for data wrappers consists in the alignment of source data with our internal data model, which is designed to contain data in its original *and* integrated representations. This internal data model is composed of two main parts:

[22] https://www.wikidata.org/wiki/Q6999
[23] https://www.wikidata.org/wiki/Q11173

1. `resource`: This dynamic field holds the original source-specific data.
2. `integration`: Used for efficient full-text searches and filtering by abstracting and transforming the original data into an integrated format. An example is provided in Listing 1.

With the `integration` field being syntactically and structurally consolidated, consuming services can formulate faceted queries over multiple data sources.

```
{
    "_integration": {
        "names": {
            "name": {
                "@value": "Bamberg",
                "@language": "en"
            },
            "variantNames": [{
                  "@value": "Bamberg",
                  "@language": "de"
                }]
            },
        "provider": "wikidata",
        "coordinates": { // ...
        },
        "id": "Q3936",
        "type": ["college town", // ...
        ],
        "descriptions": [{
                "@value": "kreisfreie Stadt in Bayern, Deutschland",
                "@language": "de"
              }],
        "url": "http://www.wikidata.org/entity/Q3936"
    }
}
```

Listing 1: Simplified example for the `integration` field of an entity.

As a final result of data preprocessing, data is saved in an intermediary JSON format. This explicit intermediary step allows us to functionally decouple wrappers from the indexing logic. New wrappers for other data sources only need to output data conforming to the constraints of our JSON format, which can then be indexed with our API.

5.3 Search and Retrieval

The Generic Search (GS) API, utilized for indexing and retrieving the data, utilizes OpenSearch as backend.[24] Individual entities can be accessed through a GET-request by specifying the index name of the data source and the entity's identifier. A key feature of the GS API is the flexible search function that supports not only the diverse requirements described in section 2, but can also be dynamically adapted towards new use cases.

The GS is highly configurable via configuration files, which allows to enable, disable and tweak different OpenSearch functionalities and provide them in terms of search profiles that can be tailored to the specific needs of use cases. A search query can be set up by combining different query types, such as full-text queries that support Lucene's Query Parser Syntax[25], as well as queries for (near-)exact matches in keyword fields. In order to support the varying search scenarios, described in section 2, different combinations of queries are used. The semi-automatic search scenario when bulk-importing catalog data has a focus on high-precision and uses exact match queries on entity identifiers and URLs among others. The user-centered manual search uses a ranking mostly based on full-text queries for high-recall results. Notably, additional configurations, such as aggregations, search suggestions, keyword highlighting, and the definition of geographical fields for utilizing distance or polygon filters can be added to the search profiles. In the user-centered search scenario for instance, keywords are aggregated to give the user an overview of the results, geographical fields are defined and query suggestions are presented.

Another important feature within the search configuration is the dynamic alteration of the result relevance score using scripts. This is currently employed to favor entities that contain links to matching entities of other providers, which we interpret as an indicator of the general relevance. These entries are preferred over others with a similar score, but cannot be potentially merged with other entities. Although experimental, early results indicate the applicability of this approach, and we will methodologically invest into this feature in the future to mitigate risks of e.g. hiding specialized entries.

The API can be configured to use different relations between entities. Presently, only `sameAs` relations between matching entities from different data providers are considered. Retrieving these related entries during the search is accomplished with one additional request on the basis of entity identifiers, resulting in a minimal performance impact. The future roadmap includes enabling the specification on how these matching entites across multiple sources can be merged and which fields from which sources to prefer when conflicts occur. Currently, the search results are returned as a list of the matching entities instead of merging them.

The options described above for customizing a search request can be specified in terms of the `POST` request body[26]. An example is given in listing 2. The request considers all

[24] https://gitlab.com/minfba/resinfra/generic-search/generic-search-api
[25] https://lucene.apache.org/core/2_9_4/queryparsersyntax.html
[26] https://c102-142.cloud.gwdg.de/adiss-gs/search/default

indices that contain geographical data (line 3). It looks for entities that contain a polygon of coordinates (line 4), and are within `10 km` of some coordinates (line 5-6). The linked entities of the given results are joined in the result list (line 7).

```
1  {
2    "query": "Bamberg",
3    "indices": ["geo"],
4    "filter": {"exists": "_integration.coordinates",
5               "_integration.countryCode": "DE"},
6    "coordinates": [49.9031, 10.8695],
7    "distance": 10000,
8    "joinLinks": true
9  }
```

Listing 2: JSON body for a search request.

6 Conclusion and Future Work

This work introduced ADISS, a search system designed to integrate multiple authority file providers. The system enables the simultaneous querying of data sources to retrieve a wide range of enriched authority data for data curation processes. Motivated by use cases from two research projects in the field of DH that currently utilize the API, we have demonstrated how our generic and configurable approach facilitates diverse methods of accessing this data.

The development of this system is ongoing, with plans to implement additional features, some of which have been briefly described in the preceding sections. Prospectively, more data providers will be integrated. Beyond the data presented in this paper, further data sources will be included such as Memorial Archives and historic place names from Geschichte Bayerns. Person records e.g. from ORCID[27] would extend the set of person entities of the GND and Wikidata, which we already started experiencing with. Overall, persons and other entity types will receive more focus in the future due to their relevance for our primary use cases.

To improve the relevance scoring of the search results, it may be advantageous to consider the usages and references of individual entities within research datasets. Entities utilized for enriching research metadata should receive higher consideration due to their employment frequency. However, any optimization might lead to unwanted side-effects like hiding entities from highly specialized providers. Due to the configurability of our solution, we expect to be able to mitigate such risks by classifying information needs and addressing them with dedicated search profiles. The introduction and fine-tuning of various search configurations

[27] https://orcid.org/

for specific use cases, and leveraging the flexibility of the system further enhances the search effectiveness. First experiences in this respect, particularly from the population of the Text+ Registry[28], are encouraging and need to be validated in comprehensive systematic evaluations in the future.

Moreover, we are currently investigating into standards, such as the Schema.org Place schema[29] or the Datacite schema[30], that can be applied to our integration model to enhance the interoperability of our system.

References

[Be06] Bennett, R.; Hengel-Dittrich, C.; O'Neill, E. T.; Tillett, B. B.: Viaf (virtual international authority file): Linking die deutsche bibliothek and library of congress name authority files. In: World library and information congress: 72nd IFLA general conference and council. 2006.

[BP11] Behrens-Neumann, R.; Pfeifer, B.: Die Gemeinsame Normdatei: Ein Kooperationsprojekt. German, Dialog mit Bibliotheken 23 (1), ed. by Nationalbibliothek, D., pp. 37–40, 2011, ISSN: 0936-1138, URL: https://katalog.fid-bbi.de/Record/185-11152.

[GH16] Gradl, T.; Henrich, A. Bibliothek Forschung und Praxis 40 (2), pp. 222–228, 2016, DOI: doi:10.1515/bfp-2016-0027, URL: https://doi.org/10.1515/bfp-2016-0027.

[Gr20] Groom, Q.; Güntsch, A.; Huybrechts, P.; Kearney, N.; Leachman, S.; Nicolson, N.; Page, R. D. M.; Shorthouse, D. P.; Thessen, A. E.; Haston, E.: People are essential to linking biodiversity data. Database 2020, baaa072, 2020, ISSN: 1758-0463, DOI: 10.1093/database/baaa072, eprint: https://academic.oup.com/database/article-pdf/doi/10.1093/database/baaa072/36143659/baaa072.pdf, URL: https://doi.org/10.1093/database/baaa072.

[Gü21] Güntsch, A.; Groom, Q.; Ernst, M.; Holetschek, J.; Plank, A.; Röpert, D.; Fichtmüller, D.; Shorthouse, D. P.; Hyam, R.; Dillen, M.; Trekels, M.; Haston, E.; Rainer, H.: A botanical demonstration of the potential of linking data using unique identifiers for people. PLOS ONE 16 (12), pp. 1–11, 2021, DOI: 10.1371/journal.pone.0261130, URL: https://doi.org/10.1371/journal.pone.0261130.

[Ha21] Hawkins, A.: Archives, linked data and the digital humanities: increasing access to digitised and born-digital archives via the semantic web. Archival Science 22, pp. 319–344, 2021, DOI: doi:10.1007/s10502-021-09381-0, URL: https://doi.org/10.1007/s10502-021-09381-0.

[HG21] Henrich, A.; Gradl, T.: Integration von Forschungsdaten. Wie können Forschungsinfrastrukturen helfen? In (Seng, E.-M.; Göttmann, F., eds.): Wesersandstein vom 16. bis 19. JahrhundertWeser Sandstone from the 16th to the 19th Century Architektur und Digital HumanitiesArchitecture and Digital Humanities. De Gruyter, Berlin, Boston, pp. 749–762, 2021, ISBN: 9783110538915, DOI: doi:10.1515/9783110538915-039, URL: https://doi.org/10.1515/9783110538915-039.

[28] https://registry.text-plus.org/

[29] https://schema.org/Place

[30] https://schema.datacite.org/meta/kernel-4.6/

[Je23] Jegan, R.; Fruth, L.; Gradl, T.; Henrich, A.: Integrating Access to Authority Data for Improved Interoperability of Research Data in the Digital Humanities. In (König-Ries, B.; Scherzinger, S.; Lehner, W.; Vossen, G., 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. Vol. P-331. LNI, Gesellschaft für Informatik e.V., pp. 829–836, 2023, DOI: 10.18420/BTW2023-54, URL: https://doi.org/10.18420/BTW2023-54.

[KLM17] Karam, N.; Lorenz, R. H.; Müller-Birn, C.: The GFBio Terminology Service. 2017.

[Me21] Menzel, S.; Schnaitter, H.; Zinck, J.; Petras, V.; Neudecker, C.; Labusch, K.; Leitner, E.; Rehm, G.: Named Entity Linking mit Wikidata und GND – Das Potenzial handkuratierter und strukturierter Datenquellen für die semantische Anreicherung von Volltexten. In (Franke-Maier, M.; Kasprzik, A.; Ledl, A.; Schürmann, H., eds.): Qualität in der Inhaltserschließung. De Gruyter Saur, Berlin, Boston, pp. 229–258, 2021, ISBN: 9783110691597, DOI: doi:10.1515/9783110691597-012, URL: https://doi.org/10.1515/9783110691597-012.

[Wi16] Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E.; Bouwman, J.; Brookes, A. J.; Clark, T.; Crosas, M.; Dillo, I.; Dumon, O.; Edmunds, S.; Evelo, C. T.; Finkers, R.; Gonzalez-Beltran, A.; Gray, A. J.; Groth, P.; Goble, C.; Grethe, J. S.; Heringa, J.; 't Hoen, P. A.; Hooft, R.; Kuhn, T.; Kok, R.; Kok, J.; Lusher, S. J.; Martone, M. E.; Mons, A.; Packer, A. L.; Persson, B.; Rocca-Serra, P.; Roos, M.; van Schaik, R.; Sansone, S.-A.; Schultes, E.; Sengstag, T.; Slater, T.; Strawn, G.; Swertz, M. A.; Thompson, M.; van der Lei, J.; van Mulligen, E.; Velterop, J.; Waagmeester, A.; Wittenburg, P.; Wolstencroft, K.; Zhao, J.; Mons, B.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3 (1), p. 160018, 2016, ISSN: 2052-4463, DOI: 10.1038/sdata.2016.18, URL: https://doi.org/10.1038/sdata.2016.18.

[Zh22] Zhao, F.: A systematic review of Wikidata in Digital Humanities projects. Digital Scholarship in the Humanities 38 (2), pp. 852–874, 2022, ISSN: 2055-7671, DOI: 10.1093/llc/fqac083, eprint: https://academic.oup.com/dsh/article-pdf/38/2/852/50488385/fqac083.pdf, URL: https://doi.org/10.1093/llc/fqac083.

Workshop on Data Engineering for Data Science

Second Workshop on Data Engineering for Data Science

Marina Tropmann-Frick [1], Tanja Auge [2] und Sheeba Samuel [3]

Data engineering serves as the foundation for meaningful data analysis, encompassing critical processes such as data collection, integration, and metadata management. These tasks are essential for ensuring data quality, reliability, and accessibility, yet they frequently constitute the most time-consuming aspects of data science workflows. Optimizing these processes is essential to improve both efficiency and analytical accuracy.

The one-day workshop address key challenges and advancements in data engineering, with a focus on best practices and emerging methodologies. Topics of discussion include data ingestion, cleaning, integration, and storage, alongside strategies for metadata management and data provenance assurance. By convening researchers and practitioners, this workshop aims to foster discourse on improving data workflows and infrastructure, ultimately contributing to the advancement of the data science discipline.

The scope of the workshop includes, but is not limited to:

- Interplay between data engineering and data science.
- Dedicated database and dataflow architectures.
- Managing data and event streams.
- Scalable data processing in data science.
- Managing metadata in data science projects.
- Data provenance in data science projects.
- Reproducibility and replicability of data analysis.
- Knowledge discovery in data science applications.
- Data and information visualization.
- Data and information flow engineering and management.
- Privacy preserving data, information, and information systems.
- Impact of Data Quality and Data Preprocessing on the Fairness of ML Predictions.

[1] Hamburg University of Applied Sciences, Germany,
marina.tropmann-frick@haw-hamburg.de, https://orcid.org/0000-0003-1623-5309
[2] University of Regensburg , Germany,
tanja.auge@informatik.uni-regensburg.de, https://orcid.org/0009-0006-2150-9713
[3] Chemnitz University of Technology, Germany,
sheeba.samuel@informatik.tu-chemnitz.de, https://orcid.org/0000-0002-7981-8504

doi:10.18420/BTW2025-124

- Data Management in Machine Learning Applications.
- Definition, Execution and Optimization of Complex Data Science Pipelines.
- Dataset preprocessing and data integration for Data Science projects.
- Handling of non-i.i.d. data for Data Science projects like graphs.
- Development of dedicated benchmarks for evaluating data engineering solutions.
- Reports about data science projects.

The workshop is an initiative of the DBIS Working Group "Data Engineering for Data Science".

Programm

Given the scope of the workshop, we are pleased to announce a great program. The workshop starts with a keynote by Laura Koesten (University of Vienna). For the technical program we could accept four full papers, three short papers and one abstract, organized in two sessions. The first talk by Restat et al. (University of Hagen) presents ALPINE, a language for describing data cleaning pipelines. The next two talks focused on eye-tracking data. While Hausler et al. (University of Regensburg) implement a general, reusable storage approach via a graph schema, Landes et al. (University of Regensburg, Hochschule Neu-Ulm) highlight the impact of preprocessing on classification results of such data. The session concludes with Heinz et al. (OTH Regensburg) presenting SQLinked, a hybrid approach for local and remote program execution.

In Session 2, Risis et al. (DFKI & Technical University of Darmstadt) talk about answering complex table questions over tabular data lakes, followed by Stangl et al. (Friedrich-Alexander-Universität Erlangen-Nürnberg & Ostbayerische Technische Hochschule Amberg-Weiden) who present the Kosmosis use case of crypto rug pull prevention by an incrementally constructed knowledge graph. After that, Brinkmann et al. (University of Mannheim) point out self-refinement strategies for LLM-based product attribute value extraction. Finally, Schüle (University of Bamberg) presented his work on higher-order SQL lambda functions, originally published at ICDE 2024. In summary, the workshop offers a variety of topics, each with a distinctive character.

Last but not least, we would like to thank everyone who contributed to this workshop, especially the authors, the reviewers, the BTW team, and all the participants.

Workshop Organizers

Tanja Auge (University of Regensburg)
Marina Tropmann-Frick (Hamburg University of Applied Sciences)
Sheeba Samuel (Chemnitz University of Technology)

Program Committee

Ahmed Al-Ghezi (Goethe University Frankfurt)
Andreas Thor (HTWK Leipzig)
Annett Ungethüm (University of Hamburg)
Benjamin Hättasch (TU Darmstadt)
Bostjan Brumen (Maribo University)
Kerstin Schneider (Harz University of Applied Science)
Martin Schultz (HAW Hamburg)
Matthias Boehm (Graz University of Technology)
Meike Klettke (University of Regensburg)
Pratyush Agnihotri (TU Darmstadt)
Richard Lenz (FAU, Erlangen-Nürnberg)
Ralf Schenkel (Trier University)
Thomas Seidl (LMU)

ALPINE: Abstract Language for Pipeline Integration and Execution

Valerie Restat [1] und Uta Störl [1]

Abstract: When working with data, it is essential to ensure data quality and clean data of errors. This is usually done with a data cleaning pipeline. The execution of such a pipeline is possible with a variety of tools. However, to increase reusability, there should be a way to describe pipelines independently of technology. So far, there is no such technology-independent description. This paper therefore presents ALPINE, a language for describing data cleaning pipelines. This abstracts from the concrete implementation. After introducing the individual components, the usage is illustrated using a running example.

Keywords: data cleaning pipeline, description language, pipeline profile, data quality

1 Introduction

Data almost always contains errors such as missing or inconsistent values. One of the most important steps when working with data is therefore data cleaning, which is intended to ensure data quality. This is usually done with the help of a *data cleaning pipeline*, in which various operators are executed that, for example, replace missing values or standardize inconsistent values.

There are many different tools and frameworks for creating such a data cleaning pipeline (e.g. Python scikit-learn[2], Airflow[3] or Dagster[4]). The choice of technology depends on various factors, such as the personal experience of the data engineer or company constraints. To ensure reusability, it should be possible to describe a data cleaning pipeline in a technology-independent way [RW20]. This also needs to be considered when designing the best possible pipeline, which is the subject of current research (see [SKB23, KW19, Ne21, RKS24]). So far, however, there is no universal description of a data cleaning pipeline that abstracts from the specific implementation.

The contribution of this paper is thus the definition of a description language for data cleaning pipeline in form of a pipeline profile. The use of this language enables the implementation and execution of pipelines in various technologies. That is why we have named it ALPINE (Abstract Language for Pipeline Integration and Execution). Transferability of technologies

[1] University of Hagen, Chair of Databases and Information Systems, Universitätsstr. 1, 58097 Hagen, Germany, valerie.restat@fernuni-hagen.de, https://orcid.org/0000-0002-5960-5886; uta.stoerl@fernuni-hagen.de, https://orcid.org/0000-0003-2771-142X

[2] https://scikit-learn.org/stable/modules/preprocessing.html

[3] https://airflow.apache.org/

[4] https://dagster.io/

doi:10.18420/BTW2025-125

is particularly important for scenarios in which the pipelines are executed repeatedly and are in operation for a long time (in contrast to ad hoc analyses). The choice of the right technology is also aided, as the abstraction simplifies the evaluation of different options.

The remainder of this paper is structured as follows. Related work is presented in Section 2. Section 3 introduces the proposed description language and demonstrates its use in a running example. The results of this paper are summarized in Section 4.

2 State of the Art

To the best of our knowledge, there is no language to describe data engineering pipelines. Other works describe the pipeline, if at all, either only conceptually [SKB23, Ab16, AHS23] or in concrete technology [Gr21].

However, in other domains, de facto standards have become established. There are various description languages for the specification of different processes. An example is the Common Workflow Language (CWL)[5] [Am16]. Using CWL, command line tools can be described and linked to workflows. The description can be provided in YAML or JSON. In the area of machine learning, ONNX[6] exists. It seeks to provide a common language that any machine learning system can use to define its models. MLSchema[7] is also available for machine learning workflows. It contains various classes and properties that can be used to represent and exchange information about data mining and machine learning algorithms, datasets and experiments. For modeling business processes, BPMN [CT12] has been established as the standard. Process diagrams can be created using a graphical notation. However, these languages cannot be used for data engineering pipelines without further consideration. As described, a generally applicable description language for a data engineering pipeline could not be found and is thus the contribution of this paper.

3 ALPINE: Definition of Pipeline Profiles

As described, ALPINE can be used to abstractly describe a pipeline in order to isolate it from the realization in a specific technology. An implementation can then be carried out using *adapters*, which is explained in more detail in the following subsections.

JSON was selected for the profile format since it is a compact format for data exchange in readable form. In addition, JSON Schema provides a description language for the structure of JSON data. This offers an advantage over the YAML format, for example.

5 https://www.commonwl.org/user_guide/introduction/quick-start.html
6 https://onnx.ai/onnx/intro/concepts.html
7 https://ml-schema.github.io/documentation/ML%20Schema.html

3.1 Running Example

In the following, the usage of the description language is introduced using an example. For this purpose, a pipeline is built for a *cheese data set* (Table 1). It contains information about cheese: `CheeseID`, `CheeseName`, `CheeseType`, `CheeseAge`, and `PricePerKilo`, which describe the respective ID, name, type, age, and price per kilo for each cheese.

Tab. 1: *Cheese Data Set*. Errors are marked in red.

CheeseID	CheeseName	CheeseType	CheeseAge	PricePerKilo
1	Mozzarella	Dried	8	
2	Pecorino	Flavoured	10	22.22
3	Brie		21	155.39
4	Roquefort	Grated	111	19.37
5	Emmental	Flavoured	14	27.70
6	Camembert		5	19.32
7	Raclette	Soft	4	34.80
8	Kasseri	Grated	15	339.50
9	Burrata	Hard	2	
10	Appenzeller	Flavoured	8	38.83

The data set contains errors that were detected in a previous step. In Table 1, the errors are marked in red. The following errors are present:

- *Missing Values* can be found in `CheeseType` and `PricePerKilo`
- *Outliers* can be found in `CheeseAge` and `PricePerKilo`

In the following, a pipeline will be built to clean these errors. To this end, the components of the corresponding pipeline profile are described step by step. It is then shown how this can be used to clean up the data set in Python using a corresponding adapter.

3.2 Components

As already mentioned, the pipeline profile is created in JSON format. The individual elements are described below and illustrated using the example of the *cheese data set*. The associated JSON Schema to validate the pipeline profile can be found in Appendix B.

general General information is specified here, such as the ID of the pipeline, the time at which it was created and the associated data set. This is required, for example, to be able to compare different profiles with each other.

```
"general": {
    "pipeline_id": "P1",
    "created": "2024-11-11 11:11:11",
    "dataset": "cheese.csv" }
```

Besides the general information, the profile contains an array of JSON objects named *operators*. These objects contain the following elements:

id Each operator has an *ID* so that it can be uniquely identified. Among other things, this is important for specifying the position within the pipeline at which the operators must be executed (see description of *position*). It can be indicated as follows:

```
"id": "O1"
```

properties All *properties* that the operator processes are specified here in an array. In the case of *cheese data set*, the properties are the *attributes* of the data set. However, attributes are strongly associated with relational data. We intentionally want to apply the more general term *properties* in order not to exclude other data models explicitly. This follows the definition from [KH22], which defines property as a unifying term.

An operator that replaces the missing values in `CheeseType` would contain the following information, for example:

```
"properties": ["CheeseType"]
```

algorithm class The *algorithm class* describes which type of error the operator handles. In the case of the *cheese data set*, there are missing values and outliers as described. For this reason, operators of the algorithm classes *missing value imputation* and *outlier handling* are required.

If an operator imputes missing values, for example, this can be specified as follows:

```
"algorithm_class": "missing-value-imputation"
```

position There are operators in a pipeline that must be executed in a certain order. For example, if we want to replace the missing values in `PricePerKilo` with the mean value, we should first clean up the outliers, as these would otherwise distort the mean value. For this reason, the *position* of an operator in the pipeline must be specified. However, it is important to note that not all operators have to be executed in a fixed order. For example, the correction of missing values in `CheeseType` is independent of the correction of outliers in `CheeseAge`. This can therefore be executed in parallel or in any order.

If, for example, the operator with the ID O_2 is to be executed before the operator with the ID O_3, but after this with the ID O_1 this can be specified as follows:

```
"position": {"before": ["O3"], "after": ["O1"]}
```

Even if one direction (e.g. only `after`) would be sufficient, we have decided that both directions can be specified. The unidirectional notation can be selected to reduce errors. Users who prefer a higher readability can choose the bidirectional notation.

If an operator can be executed at any time, the element position can be omitted or empty arrays can be passed:

```
"position": {"before": [], "after": []}
```

parameters Most operators have *parameters* that specify the execution in more detail. For example, missing values can be replaced with the mean or the median, the entire row or the entire property can be deleted, among other possibilities. For this reason, the parameters of the operator are defined here within a JSON object. The specification is made via key-value pairs. If, for example, the missing values in `PricePerKilo` are to be replaced with the mean value, this can be specified as follows:

```
"parameters": {"strategy": "replace", "fill_value": "mean"}
```

3.3 Application Example

After the components were introduced in the previous section, this section presents a concrete example of how to clean the errors in the *cheese data set* and provides an overview of the entire process.

The pipeline used to clean the *cheese data set* is shown in Figure 1.

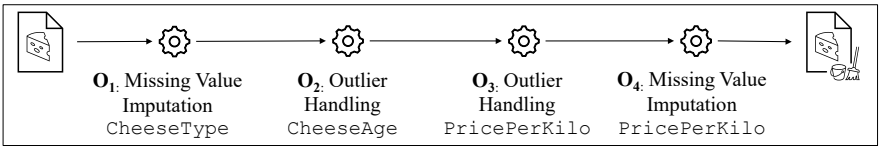


Fig. 1: Overview of the pipeline to clean the *Cheese Data Set*

It consists of the following operators:

- O_1: Imputes *Missing Values* in `CheeseType`: This is categorical data. It is decided to replace the missing values with a new category "Unknown".
- O_2: Handles *Outliers* in `CheeseAge`: This is numerical data. We decide to delete the row with the error. As this is only one row, we do not lose much information.
- O_3: Handles *Outliers* in `PricePerKilo`. These are also numerical values that are replaced by the median value.
- O_4: Imputes *Missing Values* in `PricePerKilo`: These are also numerical values that are replaced by the mean value.

Even though the figure shows the pipeline operators in a sequential order for the purpose of clear presentation, this only plays a role for two operators: O_3 and O_4. As the decision was made to replace the missing values with the mean value for O_4, the outliers should be corrected first. In this way, the mean value is not distorted. All other operators can be

executed at any point in the pipeline. This can also be seen from the element *position* in the associated pipeline profile, which is shown in Appendix A.

This pipeline profile can now be used to create a concrete data cleaning pipeline using adapters. An overview of the process can be seen in Figure 2. In our example, a Python adapter is used. The corresponding prototypical code as well as the data and profiles can be found on Gitlab[8]. The example can be tested within a Jupyter Notebook. The associated Python adapter operates on pandas DataFrames and executes the described cleaning steps. Other adapters could, for example, generate a DAG for Apache Airflow.

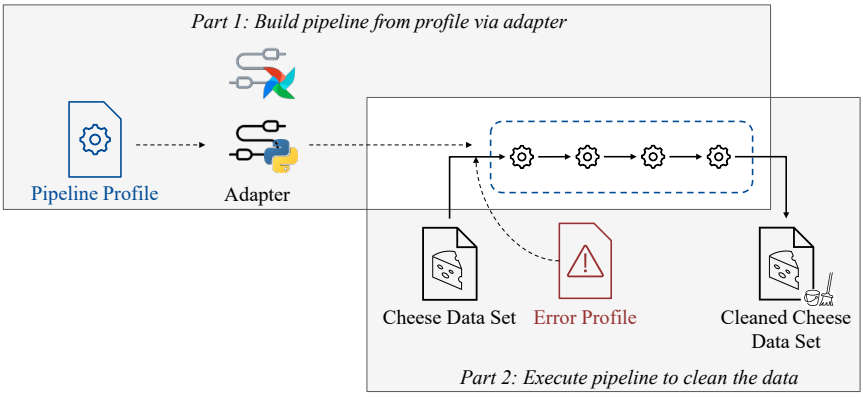


Fig. 2: Overview of generating a concrete pipeline using the pipeline profile

Once the specific pipeline has been created using the adapter, only the data set to be cleaned and an *error profile* are required. In our example, we use an error profile that was created with a tool called *CheDDaR* [RKS23]. CheDDaR provides a comprehensive JSON report on errors in a data set. It is primarily intended for evaluating data quality, but can also be used for the purpose of error detection. Other tools or approaches could also be used; the important thing is that the error type (e.g. *missing values*) and the index of the corresponding errors are available so that it is known which data points need to be cleaned and how.

Once the pipeline has been completed, the cleaned data set is available (see Appendix C).

4 Conclusion and Future Work

In this paper, with ALPINE, a description language for pipeline profiles was described. Using this, data cleaning pipelines can be described in a technology-independent way and thus abstracted from the concrete implementation. This promotes the reusability. It may also be used as an exchange format between different systems. The creation of the best possible pipeline can be carried out independently of technology and can thus be used for different environments.

[8] https://gitlab.com/d6745/alpine

In future research, we want to implement further adapters. We also want to analyze how the pipeline profile can be automatically derived from a specific implementation. This would make it possible to transfer pipelines from one technology to another.

Acknowledgement

Thanks to André Conrad for his help in providing the code and the docker and git tips!

Bibliography

[Ab16] Abedjan, Ziawasch; Chu, Xu; Deng, Dong; Fernandez, Raul Castro; Ilyas, Ihab F.; Ouzzani, Mourad; Papotti, Paolo; Stonebraker, Michael; Tang, Nan: Detecting Data Errors: Where are we and what needs to be done? Proc. VLDB Endow., 9(12):993–1004, 2016.

[AHS23] Abdelaal, Mohamed; Hammacher, Christian; Schöning, Harald: REIN: A Comprehensive Benchmark Framework for Data Cleaning Methods in ML Pipelines. In: EDBT. OpenProceedings.org, pp. 499–511, 2023.

[Am16] Amstutz, Peter; Crusoe, Michael R.; Nebojša Tijanić; Chapman, Brad; Chilton, John; Heuer, Michael; Kartashov, Andrey; Leehr, Dan; Ménager, Hervé; Nedeljkovich, Maya; Scales, Matt; Soiland-Reyes, Stian; Stojanovic, Luka: Common Workflow Language, v1.0, 2016.

[CT12] Chinosi, Michele; Trombetta, Alberto: BPMN: An introduction to the standard. Comput. Stand. Interfaces, 34(1):124–134, 2012.

[Gr21] Grafberger, Stefan; Guha, Shubha; Stoyanovich, Julia; Schelter, Sebastian: MLINSPECT: A Data Distribution Debugger for Machine Learning Pipelines. In: SIGMOD Conference. ACM, pp. 2736–2739, 2021.

[KH22] Koupil, Pavel; Holubová, Irena: A unified representation and transformation of multi-model data using category theory. J. Big Data, 9(1):61, 2022.

[KW19] Krishnan, Sanjay; Wu, Eugene: AlphaClean: Automatic Generation of Data Cleaning Pipelines. CoRR, abs/1904.11827, 2019.

[Ne21] Neutatz, Felix; Chen, Binger; Abedjan, Ziawasch; Wu, Eugene: From Cleaning before ML to Cleaning for ML. IEEE Data Eng. Bull., 44(1):24–41, 2021.

[RKS23] Restat, Valerie; Klettke, Meike; Störl, Uta: FAIR is not enough - A Metrics Framework to ensure Data Quality through Data Preparation. In: BTW. volume P-331 of LNI. GI, pp. 917–929, 2023.

[RKS24] Restat, Valerie; Klettke, Meike; Störl, Uta: Towards an End-to-End Data Quality Optimizer. In: ICDEW. IEEE, pp. 262–266, 2024.

[RW20] Romero, Oscar; Wrembel, Robert: Data Engineering for Data Science: Two Sides of the Same Coin. In: DaWaK. volume 12393 of Lecture Notes in Computer Science. Springer, pp. 157–166, 2020.

[SKB23] Siddiqi, Shafaq; Kern, Roman; Boehm, Matthias: SAGA: A Scalable Framework for Optimizing Data Cleaning Pipelines for Machine Learning Applications. Proc. ACM Manag. Data, 1(3):218:1–218:26, 2023.

A Pipeline Profile for Cheese Data Set


```
{
  "pipeline": {
    "general": {
      "pipeline_id": "P1",
      "created": "2024-11-11 11:11:11",
      "dataset": "cheese.csv"
    },
    "operators": [
      {
        "id": "O1",
        "properties": ["CheeseType"],
        "algorithm_class": "missing-value-imputation",
        "position": {},
        "parameters": {"strategy": "replace", "fill_value": "Unknown"}
      },
      {
        "id": "O2",
        "properties": ["CheeseAge"],
        "algorithm_class": "outlier-handling",
        "position": {},
        "parameters": {"strategy": "delete_rows"}
      },
      {
        "id": "O3",
        "properties": ["PricePerKilo"],
        "algorithm_class": "outlier-handling",
        "position": {"before": ["O4"]},
        "parameters": {"strategy": "replace","fill_value": "median"}
      },
      {
        "id": "O4",
        "properties": ["PricePerKilo"],
        "algorithm_class": "missing-value-imputation",
        "position": {"after": ["O3"]},
        "parameters": {"strategy": "replace","fill_value": "mean"}
      }
    ]
  }
}
```

B JSON Schema for Pipeline Profile

```
{
  "$schema": "http://json-schema.org/draft2020-12/schema#",
  "type": "object",
  "properties": {
    "pipeline": {
      "type": "object",
      "properties": {
        "general": {
          "type": "object",
          "properties": {
            "pipeline_id": {"type": "string"},
            "created": {"type": "string"},
            "dataset": {"type": "string"}
          },
          "required": [
            "pipeline_id",
            "created",
            "dataset"
          ]
        },
        "operators": {
          "type": "array",
          "items": {
              "type": "object",
              "properties": {
                "id": {"type": "string"},
                "properties": {
                  "type": "array",
                  "items": {"type": "string"}
                },
                "algorithm_class": {"type": "string"},
                "position": {
                  "type": "object",
                  "properties": {
                    "before": {
                      "type": "array",
                      "items": {"type": "string"}
                    },
                    "after": {
                      "type": "array",
                      "items": {"type": "string"}
```

```
                    }
                  }
                },
                "parameters": {"type": "object"}
              },
              "required": [
                "id",
                "properties",
                "algorithm_class",
                "parameters"
              ]
            }
          }
        },
        "required": [
          "general",
          "operators"
        ]
      }
    },
    "required": [
      "pipeline"
    ]
}
```

C Cleaned Cheese Data Set

Tab. 2: Cleaned *Cheese Data Set*

CheeseID	CheeseName	CheeseType	CheeseAge	PricePerKilo
1	Mozzarella	Dried	8	28.32
2	Pecorino	Flavoured	10	22.22
3	Brie	Unknown	21	27.70
5	Emmental	Flavoured	14	27.70
6	Camembert	Unknown	5	19.32
7	Raclette	Soft	4	34.80
8	Kasseri	Grated	15	27.70
9	Burrata	Hard	2	28.32
10	Appenzeller	Flavoured	8	38.83

SeeME: A General, Reusable Graph Schema for Data Preprocessing of Eye-Tracking Data

Dominique Hausler [1], Jennifer Landes [1], and Meike Klettke [1]

Abstract: To track eye movement over time and to gain information about points of interest through fixation data, eye-tracking is used in a wide range of fields. In this paper, we present a general, reusable approach to store eye-tracking data and to realize data preprocessing tasks in-database. To achieve this, a graph databases schema for any eye-tracking data, consisting of 1) a time series data level and 2) a meta level is developed. Follow-up experiments or additional data like demographic data can easily be integrated into the meta level of the general schema. We use Neo4j to implement this general graph schema. To prepare the time series data for machine learning tasks, we additionally present a modular in-graph-database preprocessing pipeline, empowering researchers to either compare different operators or select the best fitting one. For each preprocessing step Cypher code for at least two preprocessing algorithms for time series data are at hand.

Keywords: Data Preprocessing, Graph Database, Neo4j, Graph Schema, Eye-Tracking Data, Time Series Data

1 Introduction

Eye-tracking data are used in a wide variety of research fields such as psychology, usability testing, comprehension of foreign languages, or analyzing the differences between novices and amateurs for programming tasks. Eye-tracking results are commonly delivered by the system as CSV files following a specific schema. In order to efficiently store the gained data and extend it by additional experiment related data (see Figure 1), databases are beneficial over the use of CSV files. This ensures (1) easy access to the data, (2) data manipulation in a single store, and (3) cooperative work. During all eye-tracking experiments, data like gazes, saccades, timestamps, and fixations over a specific time period are recorded. Therefore, eye-tracking data are an example for highly interconnected time series data. Accordingly, we suggest the usage of a graph database to conduct data preprocessing. Building the base for further analysis tasks with machine learning models or classification. As a result, the following two **R**esearch **Q**uestions (RQs) emerge:

- RQ1: How to adequately model eye-tracking data in a graph database?
- RQ2: How to realize data preprocessing in-graph-database?

1 University of Regensburg, Data Engineering Group, Bajuwarenstrasse 4, 93053 Regensburg, Germany,
dominique.hausler@ur.de, https://orcid.org/0009-0004-2381-133X;
jennifer.landes@stud.uni-regensburg.de, https://orcid.org/0009-0003-1914-598X;
meike.klettke@ur.de, https://orcid.org/0000-0003-0551-8389

doi:10.18420/BTW2025-126

In the context of RQ1, we present a general graph schema, specialized for modeling eye-tracking data. This is advantageous for adequately handling the data gained from any eye-tracking experiment. **Out of three reasons, a graph database is recommended**: (i) Even though graph databases are schema-less, an implicit structure can be derived from the data. (ii) Graph databases allow an easy and fast recognition of connections e.g., between the gaze data of test persons. (iii) Paths between predefined points of interest can be illustrated in an intuitive way, e.g., revisits of gazes to objects needed for navigation tasks. (iv) Gaze paths can be efficiently modeled and contextual information can be easily integrated.

In order to answer RQ2, we present an in-graph-database approach, incorporating a human-in-the-loop approach. Our implementation, called SeeME, allows researchers to oversee and interact with each step of the preprocessing pipeline (highlighted in yellow in Figure 1). On the one hand, this ensures data quality and context-specific adjustments, on the other hand further analysis can be done after utilizing the preprocessing presented in this paper. The preprocessing pipeline includes feature selection, data preparation and normalization. All steps are implemented in-graph-database, meaning within a Neo4j graph database, to reduce data movement and to minimize complexity by using a single data model. This eliminates the need to manage multiple CSV files across variouse locations. Furthermore, modularity is offered, since at least two operators are implemented for each preprocessing step. This enables researchers to flexibly combine preprocessing steps and run different pipelines for result comparison. A quick data overview is provided through the visualization of connections upon querying the Neo4j database.

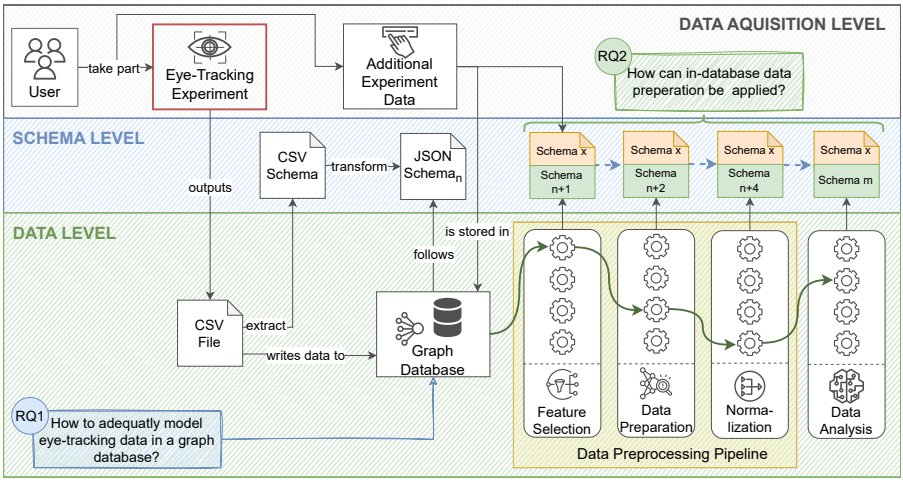


Fig. 1: From Data Acquisition to Classification

Problem Description. Figure 1 shows the complete workflow from the experiment to data analysis. For RQ1, the focus lies on the schema of the eye-tracking data. Here we will show how the CSV file can be transformed to the proposed graph schema within Neo4j. Afterwards,

Cypher code for different operators of the data preprocessing pipeline is presented in the course of answering RQ2. Both RQs include an in-graph-database implementation with Neo4j to reduce data movement and consequently, improve performance. Two timelines can be identified in our research: (1) the time series data gained through the eye-tracking experiment and (2) the schema changes of the time series data (green blocks at schema level in Figure 1), while the metadata schema (orange blocks) remain (mostly) unchanged while being adaptable and extendable. The metadata refers to additional experiment data such as pre- or post-test results or demographic data.

Contribution. Until now, researchers had to deal with multiple CSV files in different locations. We aim to ease access to the data by using a single database in which the dataset can be extended by additional experiments. The novelty lies in the following points:

- A general, reusable graph schema for eye-tracking data to perform data preprocessing.
- An in-graph-database preprocessing pipeline for eye-tracking data with a human-in-the-loop approach.

Structure. First, relevant literature is analyzed in Section 2, followed by the general, reusable graph schema, to store any eye-tracking data (Section 3). Afterwards, the precise in-graph-database implementation of our preprocessing pipeline is presented in Section 4, continuing with the conclusion.

2 Related Work

In this section, topic-related work is presented. To answer RQ1, work on graph schema is discussed. Afterwards, we analyze if eye-tracking data is usually stored in databases. Due to eye-tracking data resembling time series data, approaches to store such are presented, followed by preprocessing approaches for multivariate time series data (MVTS). This is important because it involves analyzing multiple variables over time to uncover patterns and interactions, such as those in eye-tracking data. In the context of RQ2, we continue with work on preprocessing for eye-tracking data as well as in-database preprocessing.

Graph Schema. Schema extraction from semi-structured data is an ongoing research topic, for NoSQL databases suggested in [Kl16; SK22] and for graph databases the importance of a schema was addressed in [An21; An23] and PG-Triggers developed in [Ga23]. Moreover, Neo4j itself offers the option of using constraints [Ne24; PVK17]. Our work concentrates on a general schema for any eye-tracking experiment in the context of data preprocessing.

Eye-Tracking Datasets vs. the Usage of Databases. Only a few eye-tracking experiments offer what they call a *database*. This is no database in the sense of a database system but a dataset containing eye-tracking data and possibly additional files to reconstruct the performed analysis. Currently, storing eye-tracking data in any database is a gap, not yet researched. [En10] offers an image dataset, matlab files to reconstruct the analysis and text

files with general information. In [HEB12] the data are available in Bayer format. In [CW19; Gu20; KKM14], a large open source dataset including additional experiment data like demographics or statistics are presented and used for data science analysis. [No24] presents a dataset with 360° videos. None of them investigates the development of an ideal database schema nor makes use of a database system or perform data preprocessing in-database.

Handling Time Series Data. Figure 2 shows different variants to model time series data. All elements representing time such as timestamps or duration, are highlighted in green.

- Case A: Timestamp nodes are connected with an event E, while arrows demonstrate the timeline. This is deficient if multiple events occur at identical timestamps [Ho20].
- Case B: The dashed arrows indicate entity changes while the arrow length represents the duration of each version in the temporal graph. The header shows when instance changes occur, whereas in the rows represent the number of versions [GBP18].
- Case C: A classical way to model time is a tree structure [Ho20; Ne14]. This is done by breaking time down into smaller sub-units.
- Cases D and E: The Entity-State (ES) Model uses entities E and states S – similar to events in Case A. In contrast to A, more than one state S can be connected to an entity E. The arrows illustrate the order in which states occurred. Moreover, it allows adding more relationships (=arrows) if a state is rolled back. Two arrows indicate the current status [h-24; He08].
- Case E: A further development of the ES-Model. Here the R node holds all information about how the relationships of E changed [h-24].

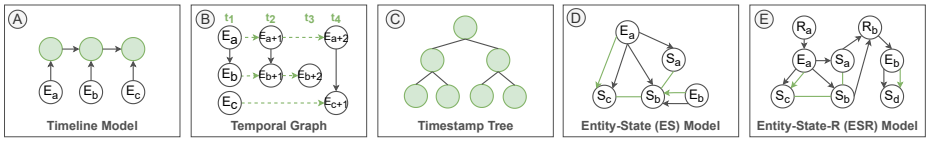


Fig. 2: Comparison of Timeline Modeling Approaches (all green elements show where and how time is represented) – selected and combined from [GBP18; h-24; He08; Ho20; Ne14]

Multivariate Time Series Data (MVTS). [Sc19] presents a framework for preprocessing heterogeneous MVTS and event data with an emphasis on preserving spatial and structural relationships. [Be19] propose a visual-interactive system for preprocessing MVTS, focusing on six key tasks: pipeline creation, routine assessment, parameter tuning, dimension-specific analysis, uncertainty analysis, and global overview. It supports a variety of preprocessing tasks while enabling iterative, user-driven adjustments. [Ah20] introduce an open-source Python package for preprocessing MVTS datasets. In contrast to these works, we focus on an in-graph-database implementation with a human-in-the-loop approach to enable researchers to choose the operators they need at each preprocessing step.

Data Preprocessing in the Context of Eye-Tracking. Preprocessing is crucial to ensure the reliability of eye-tracking data in machine learning and behavioral studies. Key steps include

feature selection, data cleaning, and handling outliers and missing values. To improve data comparability and model performance, normalization and standardization are used [LKK24]. Noise removal and outlier handling are essential to reduce misclassification, employing methods such as statistical analysis [Hu06] or distance-based approaches [AP05]. Missing values can be imputed using techniques like Zhang et al.'s NIIA, which leverages both complete and incomplete data [ZJZ11]. Discretization [DKS95] and feature selection help reduce dimensionality, enhancing algorithms such as k-NN [AKV19]. For specific applications, [Kh23] demonstrate a generalized approach to improve predictions when using aggregated eye-tracking data to detect autism spectrum disorder. We focus on modularity, meaning we will present a number of operators that can be freely combined. Moreover, we illustrate how these operators can be implemented in Neo4j's Cypher.

Data Preprocessing In-Database. There are some approaches suggesting the integration of machine learning methods in relational databases e.g., [Sc23]. In [Ba20] SQL is used to perform e.g., data cleaning as well as preprocessing. Data preprocessing during data imputation for relational databases was also presented in [PN24a]. To the best of our knowledge, there are no similar approaches for graph databases. In this article, we present an in-database approach, using the popular graph database Neo4j. The advantages are obviously that (i) no data need to be moved between systems and (ii) that user can simply perform all steps in one system.

3 Approach: General, Reusable Graph Schema

This section focuses on answering RQ1 How to adequately model eye-tracking data in a graph database? First, the schema of a CSV file is described as in [Sh05], followed by the schema of a graph database. To retrieve the proclaimed graph schema, the schema of a CSV file needs to be known, as a mapping between the two is necessary. In a second step, all fixed data e.g., consistent labels are described in Table 1 and 2. Afterwards, the general, reusable graph schema for eye-tracking data is shown and described (see Figure 4).

3.1 Formal Schema Description & Mapping

Data gained from eye-tracking data is usually stored in a CSV *file* which contains an optional *header*. CRLF hereby stands for a line break. The *header* is defined as a set of *names* separated by a *comma*, while *records* are defined as set of *fields*, containing specific values (*escaped* or *non-escaped*) [Sh05].

In order to use the data, it now needs to be mapped to a graph G, defined as a set of vertices V and edges E. A function f is used to extract the schema S_G from the graph G. A node is defined as a set of labels L, due to the possibility of multi-labeling in property graphs, and a set of properties P. A property P is defined as tuple, containing a property key n_p and a

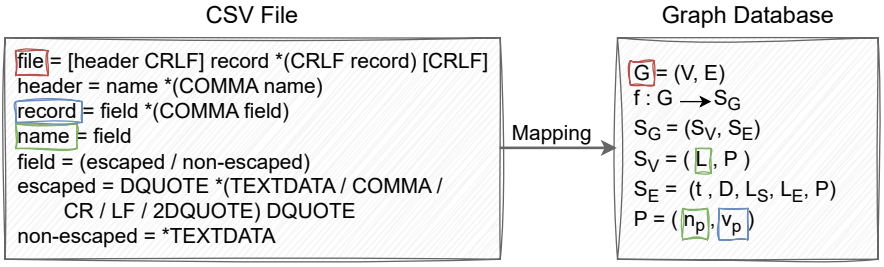


Fig. 3: Formalization of the Mapping Between the CSV File [Sh05] and the Graph Database

value v_p. Properties are so-called key-value pairs. An edge contains a type t, that cannot be empty, as well as a boolean for the direction D (= out- or ingoing). Additionally, edges have a set of start L_S and end node labels L_E, to identify the associated nodes. Just like nodes, edges contain a set of properties P.

For the **mapping**, the *file* resembles the graph G, as both hold the entire, stored data. The *name*s and *records* of the CSV file are transformed to the key-value pairs in the graph database. A substring analysis of the *name*s indicates the labels L, i.e., if *name*s include the substring *Fixation* all of those, including their related *record*s, are mapped to fixation nodes. The types are used to illustrate how the data are connected to each other.

3.2 Graph Schema

Our general, reusable graph schema for eye-tracking data consists of two subgraphs for each task Task_n (see Figure 4). (1) The metadata graph G_M contains experiment related data e.g., demographic or interview data and (2) G_T is used to store the time series data, gained through the eye-tracking technology. The metadata graph G_M consists of information about the test persons (TP_n), the tasks in the experiment (Task_n) and the applied method (Method_n). The time series subgraph G_T consists of four different node entities: Timestamp V_T, Gaze V_G, Fixation V_F and Saccade nodes V_S. For each experiment n, a list of i nodes is collected and stored. In Table 1 and 2, we define each data point with $\text{Timestamp}_{n,i}$, $\text{Gaze}_{n,i}$, and so on.

In order to acquire a complete schema, the nodes in Table 1 need to be connected through associated *edges* which are illustrated in Table 2. Therefore, that the metadata graph G_M is adaptable and extendable, depending on the additional experimental data (also see Figure 1), the edges and nodes for the metadata level are exemplarily. In our case, test person nodes V_P are connected to Task nodes V_A through the edge E_P. E_A connects a Task V_A with a Method node V_M. Table 2 shows that none of the edges in our presented schema contains properties. We assume that querying of node properties is more intuitive because querying of relationships requires the definition of a more complex pattern. Each Timestamp node V_T

Subgraph	Node V	Labels (L)	Properties (P)
G_M	V_P	TP_n	Demographics
	V_A	Task_n	Type, Duration
	V_M	Method_n	Method Type
G_T	V_T	$\text{Timestamp}_{n,i}$	Pupil Size (Left/Right)
	V_G	$\text{Gaze}_{n,i}$	Coordinates (X, Y)
	V_F	$\text{Fixation}_{n,i}$	Duration, Start, End, Location (X, Y)
	V_S	$\text{Saccade}_{n,i}$	Start, End, Duration, Direction

Tab. 1: Node Schemata for Metadata G_M and Timeseries G_T.

is part of a Fixation or Saccade node. Consequently, they are connected by E_T. Moreover, each Gaze node V_G is connected to an associated Timestamp V_T. Saccade V_S and Fixation nodes V_F are connected through E_F or E_S – depending on the direction – which represents the vector movement.

Sub-graph	Edge E	Type (t)	Direction (D)	Start Labels (L_S)	End Labels (L_E)
G_M	E_P	COMPLETED	Out	TP_n	Task_n
	E_A	USED	Out	Task_n	Method_n
G_T	E_T	IS_PART_OF	Out	$\text{Timestamp}_{n,i}$	$\text{Fixation}_{n,i}$ | $\text{Saccade}_{n,i}$
	E_G	AT	Out	$\text{Gaze}_{n,i}$	$\text{Timestamp}_{n,i}$
	E_F	MOVEMENT	Out	$\text{Fixation}_{n,i}$	$\text{Saccade}_{n,i}$
	E_S	MOVEMENT	Out	$\text{Saccade}_{n,i}$	$\text{Fixation}_{n,i}$

Tab. 2: Edge Schemata for Metadata G_M and Timeseries G_T.

The flexible metadata level G_M allows the storage of additional experiment data such as demographic data or data from pre- or post-questionnaires. Therefore, the metadata schema presented in Figure 4 is exemplary, meaning this level is use-case specific and can easily be extended or changed. In contrast, the time series level G_T is static. At G_T all gaze data are connected to their timestamps and Timestamp nodes to a Fixation or Saccade node. Even though a property graph is used, only node properties are used to ease the querying process because less complex patterns need to be searched.

4 In-Graph-Database Approach

In Section 3 we answered RQ1 and developed a general, reusable graph schema for eye-tracking data. Even though differences can occur in the outputed CSV file, like the naming of features, the collected data within eye-tracking experiments is always identical (gazes, saccades and fixations). Consequently, a general schema, allowing the storage and in-graph-database processing, is beneficial over handling CSV files. Moreover, it supports comprehensive evaluation options and selective updates of values to only name a few. Due to eye-tracking data being highly interconnected all steps are performed in a graph database, in our case Neo4j.

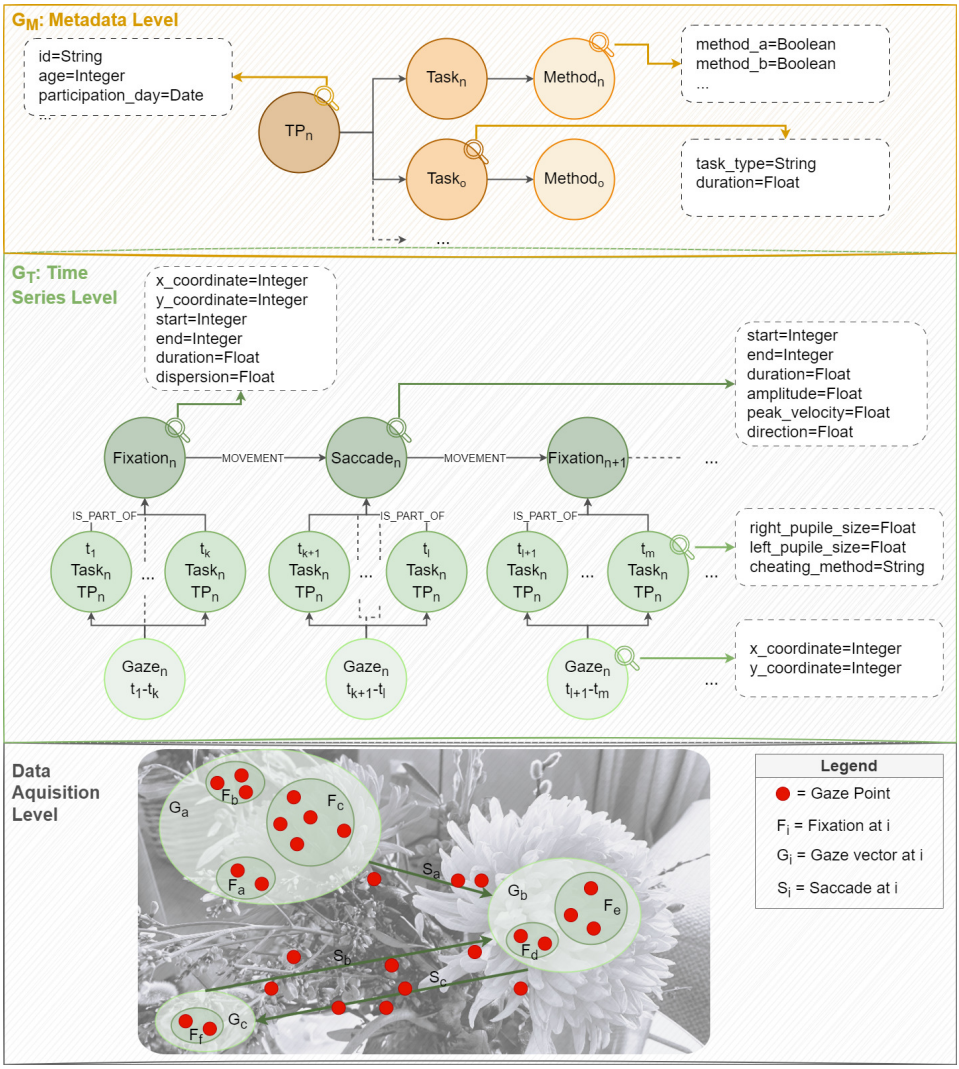


Fig. 4: General and Reusable Graph Schema for all Data Gained Through Eye-Tracking Experiments

Figure 1 illustrates the entire process from gaining eye-tracking data to the data analysis with data science techniques. In this section we focus on the precise in-graph-database implementation. We start by reading the CSV file data to Neo4j, followed by all steps within the preprocessing pipeline (highlighted in yellow in Figure 1). All of the available operators for each step can be executed in-graph-database. Our modular, block-building implementation is easy to extend and enables building an individualized pipeline by selecting the code snippets for the ideal operators of one's use case. This is done in the context of

RQ2: How to realize data preprocessing in-graph-database? The Cypher code snippets to retrieve the general graph schema together with the preprocessing pipeline is available at https://doi.org/10.5281/zenodo.14471463.

4.1 Data Preparation

When reading the gained data to Neo4j, the data first needs to be categorized into the two main categories of metadata and time series data. Time series data are defined, as all data gained through the usage of eye-tracking. Metadata refers to additional experimental data such as interviews or demographic data. Figure 1 illustrates that there are two timelines in our data: (1) First, the eye-tracking data, representing a timeline of gaze data, saccades and fixations. This means, that it measures how long someone looked at which coordinates. All steps of the preprocessing pipeline are performed to this time series data. (2) Second, the metadata schema stays stable after the schema is defined.

Most operators of each preprocessing steps only consist of value update operations, consequently, the schema stays mostly stable. Since preprocessing and schema creation are linked, the time series schema may change accordingly.

Mapping I. In order to receive the desired graph schema, the steps of the data engineering pipeline are interconnected with the operations to gain the graph schema in Figure 4. Firstly, the time series data is read to Neo4j. In this first step the node label `timestamp` is declared, meaning all data are stored in timestamp nodes in the beginning.

```
1  MATCH (n:Timestamp)
2  WHERE n.ET_ValidityLeft IS NULL OR n.
       ET_ValidityRight IS NULL
3     OR n.ET_ValidityLeft <> n.ET_ValidityRight
4     OR n.ET_ValidityLeft = n.ET_ValidityRight AND
          (n.ET_ValidityLeft = "4.0" AND n.
         ET_ValidityRight = "4.0")
5  WITH collect(n) AS nonValidNodeIdList
6  CALL apoc.nodes.get(nonValidNodeIdList)
7  YIELD node
8  DELETE node
```

List. 1: Validity Check as In-Graph-Database Operation

Data Cleaning – Validity Check. Afterwards, the data needs to be validated as demonstrated in Listing 1. If the values of the attributes *validity x* and *validity y* are both zero they are valid, otherwise an error has occurred during data acquisition (Line 2 to 4). All invalid data is collected in Line 5 and deleted from the dataset in Line 8. For performance reasons, this step is performed in the beginning, as for large datasets high-compute operations can be lowered if all invalid data are removed beforehand and not taking up additional performance.

Mapping II: Creation of Additional Labels & Hypernym Nodes. To gain the desired schema, the property key *source stimuli name*, referring to the performed task, is added as label to each timestamp node. Afterwards, the property key is deleted from the data to avoid

duplicate information. In order to create hypernym nodes i.e., nodes with a superordinate label, all property keys with the substing `Saccade` are moved to saccade nodes. The same is done for fixation and gaze. Moreover, relationships are created between the related timestamp and their associated gaze, saccade or fixation nodes as illustrated in Figure 4.

4.2 Modular Data Preprocessing Pipeline

Feature Selection. Irrelevant or redundant features that do not contribute to machine learning based eye-tracking analysis are removed. This step minimizes data complexity, focusing only on the most informative features related to gaze behavior. In our case, the name complexity of property keys is reduced to ease the understanding of the parameters. This is done to later be able to clean the data i.e., removing all irrelevant property keys (Listing 2, Line 4) starting with the substrings `Event`, `ET_` and `Interpolated` (Line 2).

```
1  MATCH (n:Timestamp)
2  WITH n, [key IN keys(n) WHERE key STARTS WITH 'Event' OR key STARTS WITH 'ET_' OR key
       STARTS WITH 'Interpolated'] AS etKeys
3  WITH collect(n) AS nodes, etKeys
4  CALL apoc.create.removeProperties(nodes, etKeys)
5  YIELD node RETURN node
```

List. 2: Deletion of Irrelevant Property Keys for ML-based Analysis as In-Graph-Database Operation

Missing Value Imputation.

We implemented four different operators, namely interpolation [KS19], deletion [ZJZ11], LOCF [ZT22] and NOCB. These can be used to perform missing value imputation for any selected node label. Missing value imputation is necessary to recognize and impute missing values to maintain data completeness.

```
1  MATCH (n:Gaze)
2  //search if any mandatory property key is empty
3  WHERE n.`Gaze X` IS NULL OR n.`Gaze Y` IS NULL OR n.`Gaze Velocity` IS NULL OR n.`Gaze
       Acceleration` IS NULL
4  WITH ID(n)-1 AS predecessorNode, ID(n) AS nodeId, ID(n)+1 AS successorNode, n, collect(n)
       AS nodes
5  // get gaze x values of following nodes
6  CALL apoc.nodes.get([successorNode])
7  YIELD node AS successorNodeList
8  // get `Gaze X` values of foregoing nodes
9  CALL apoc.nodes.get([predecessorNode])
10 YIELD node AS predecessorNodeList
11 CALL apoc.create.setProperty(nodes, 'Gaze X', (toFloat(successorNodeList.`Gaze X`) +
       toFloat(predecessorNodeList.`Gaze X`))/2)
12 YIELD node RETURN node
```

List. 3: In-Graph-Datatbase Realization of Interpolation for Gaze X

In Listing 3 interpolation for the property key `Gaze X` is illustrated. First all property keys with missing values are selected (Line 3). Next, the predecessor and successor node ids for each node with a missing value are identified in Line 4. This is necessary to calculate the mean in Line 11 for each node with a missing value and to overwrite the former null values with the results (also in Line 11). Instead of overwriting the value of the property key `Gaze X`, a new property key could be added to keep the initial recorded together with the newly calculated value. Adding another property is beneficial to maintain transparency like it is the case for Listing 5. This function can easily be replaced by other functions, resembling the individualization option. Listing 3 demonstrates that our approach is not only modular by choosing an algorithm, but also customisable by choosing a different function in Line 11 to handle missing values.

Outlier Detection. In order to detect data points that significantly deviate from the norm, an outlier detection is used. We implemented three methods. 1) A specific threshold can be set [AA23; ZWS18], 2) the z-score [Pi20] and 3) the interquartil range [VPS18]. Listing 4 demonstrates how the threshold based approach 1) can be performed in Neo4j. In order to change the threshold the value in Line 2 can be individualized or changed. In this example, all fixation nodes, holding a fixation duration below 60 milliseconds are renamed to saccade (Line 4), as they were labeled falsely.

```
1  MATCH (n:Fixation)
2  WHERE n.`Fixation Duration` < 60
3  WITH collect(n) AS outlier
4  CALL apoc.refactor.rename.label("Fixation", "Saccade", outlier)
5  YIELD batches, total, timeTaken, committedOperations
6  RETURN batches, total, timeTaken, committedOperations;
```

List. 4: Threshold Based Approach for Fixation Nodes as In-Graph-Database Operation

Normalization. Normalization is used to adjust feature scales to improve comparability across data. Consequently, supporting model performance. To address data variability two techniques are available in our implementation, in particular min-max scaling [PN24b] and the z-score standardization [Pi20].

```
1  MATCH (n:Fixation)
2  WHERE n.`Fixation Duration` IS NOT NULL
3  SET n.`Fixation Duration` = toFloat(n.`Fixation Duration`)
4  WITH min(n.`Fixation Duration`) AS minValue, max(n.`Fixation Duration`) AS maxValue
5  MATCH (n:Fixation)
6  SET n.normalized_value = (n.`Fixation Duration` - minValue) / (maxValue - minValue)
7  SET normalization_operator = 'min-max'
```

List. 5: Min-Max Normalization Algorithm as In-Graph-Database Operation with Cypher

To ensure transparency, we recommend a *parameter storage* to identify which operator was used at what preprocessing step. The property key names the preprocessing step, here *normalization*, whereas the value demonstrates the performed operator (see Listing 5 Line 7). Besides the operator, the results are stored in a property named by the operator underscore

value (e.g., normalized_value in Line 6). The advantage here is that either the results can be further used when required for an analysis like in the case of the z-score normalization or multiple operators for identical nodes can be added. Depending on the use case the property `normalization_method` is optional or an additional method node could be created to preserve the information about the utilized operators.

5 Conclusion

In the context of RQ1 we presented a general, reusable graph schema to enable adequate modeling of any eye-tracking data. This schema consists of an adaptable and extendable metadata level, containing additional experiment data (see Figure 1) and a stable time series level in which the data gained by the eye-tracking technology is stored.

In this paper, we presented an in-graph-database approach for the preprocessing of eye-tracking data to answer RQ2. By doing so, data movement is reduced and only one data model is necessary in which all preprocessing steps can be performed. A human-in-the-loop approach empowers researchers to freely choose the ideal algorithm at each preprocessing steps. For each step at least two algorithms are available in our implementation. The modular, block-building nature of our implementation is ensured by the code snippets for each operator, allowing seamless integration or extensibility. Furthermore, our implementation offers the option of testing which operators fit one's preprocessing tasks best. Due to the complexity of eye-tracking data a graph database helps to intuitively visualize such highly interconnected data. The general, reusable graph schema can be used to perform data preprocessing for the data gained by any eye-tracking experiment.

Future work will focus on evaluating the performance of this approach against alternative data models, such as relational or time series databases. Additionally, we plan on extending our approach to support other types of sensor-based time series data. We want to broaden the current implementation by further operations together with optional preprocessing steps such as filtering and smoothing. To automatize our modular pipeline, we plan on implementing a Python framework to freely choose a combination of operators. This allows an ideal preprocessing pipeline for any given use case.

6 Acknowledgements

The work of Dominique Hausler has been funded by Deutsche Forschungsgemeinschaft (German Research Foundation) – 385808805.

References

[AA23] Alhilo, T.; Al-Sakaa, A.: Handling Noisy Data in Eye-Tracking Research: Methods and Best Practices. In: 2023 International Workshop on Biomedical Applications, Technologies and Sensors (BATS). Pp. 39–44, 2023.

[Ah20] Ahmadzadeh, A.; Sinha, K.; Aydin, B.; Angryk, R. A.: MVTS-Data Toolkit: A Python package for preprocessing multivariate time series data. SoftwareX 12, p. 100518, 2020.

[AKV19] Alexandropoulos, S.-A. N.; Kotsiantis, S. B.; Vrahatis, M. N.: Data preprocessing in predictive data mining. The Knowledge Engineering Review 34, e1, 2019.

[An21] Angles, R.; Bonifati, A.; Dumbrava, S.; Fletcher, G.; Hare, K. W.; Hidders, J.; Lee, V. E.; Li, B.; Libkin, L.; Martens, W.; Murlak, F.; Perryman, J.; Savkovic, O.; Schmidt, M.; Sequeda, J. F.; Staworko, S.; Tomaszuk, D.: PG-Keys: Keys for Property Graphs. In: SIGMOD Conference. ACM, pp. 2423–2436, 2021.

[An23] Angles, R.; Bonifati, A.; Dumbrava, S.; Fletcher, G.; Green, A.; Hidders, J.; Li, B.; Libkin, L.; Marsault, V.; Martens, W.; Murlak, F.; Plantikow, S.; Savkovic, O.; Schmidt, M.; Sequeda, J.; Staworko, S.; Tomaszuk, D.; Voigt, H.; Vrgoc, D.; Wu, M.; Zivkovic, D.: PG-Schema: Schemas for Property Graphs. Proceedings of the ACM on Management of Data 1 (2), pp. 1–25, 2023.

[AP05] Angiulli, F.; Pizzuti, C.: Outlier Mining in Large High-Dimensional Data Sets. Knowledge and Data Engineering, IEEE Transactions on 17, pp. 203–215, 2005.

[Ba20] Badia, A.: SQL for Data Science - Data Cleaning, Wrangling and Analytics with Relational Databases. Springer, 2020.

[Be19] Bernard, J.; Hutter, M.; Reinemuth, H.; Pfeifer, H.; Bors, C.; Kohlhammer, J.: Visual-Interactive Preprocessing of Multivariate Time Series Data. Comput. Graph. Forum 38 (3), pp. 401–412, 2019.

[CW19] Costela, F. M.; Woods, R. L.: A free database of eye movements watching "Hollywood" videoclips. Data in Brief 25, p. 103991, 2019.

[DKS95] Dougherty, J.; Kohavi, R.; Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. In (Prieditis, A.; Russell, S., eds.): Machine Learning Proceedings 1995. Morgan Kaufmann, San Francisco (CA), pp. 194–202, 1995.

[En10] Engelke, U.; Liu, H.; Zepernick, H.; Heynderickx, I.; Maeder, A. J.: Comparing two eye-tracking databases: The effect of experimental setup and image presentation time on the creation of saliency maps. In: PCS. IEEE, pp. 282–285, 2010.

[Ga23] Gagliardi, A.; Bernasconi, A.; Martinenghi, D.; Ceri, S.: PG-Triggers: Triggers for Property Graphs. CoRR abs/2307.07354, 2023.

[GBP18] García-Domínguez, A.; Bencomo, N.; Paucar, L. H. G.: Reflecting on the past and the present with temporal graph-based models. In: MoDELS (Workshops). Vol. 2245. CEUR Workshop Proceedings, CEUR-WS.org, pp. 46–55, 2018.

[Gu20] Gupta, P.; Chugh, K.; Dhall, A.; Subramanian, R.: The eyes know it: FakeET- An Eye-tracking Database to Understand Deepfake Perception. In: ICMI. ACM, pp. 519–527, 2020.

[h-24] h-omer: Neo4j Versioner Core Documentation, Entity-State model managed by Neo4j Procedures, Accessed: 2024-08-30, 2024, URL: https://h-omer.github.io/neo4j-versioner-core/.

[He08] Heer, J.; Mackinlay, J. D.; Stolte, C.; Agrawala, M.: Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. IEEE Trans. Vis. Comput. Graph. 14 (6), pp. 1189–1196, 2008.

[HEB12] Hadizadeh, H.; Enriquez, M. J.; Bajic, I. V.: Eye-Tracking Database for a Set of Standard Video Sequences. IEEE Trans. Image Process. 21 (2), pp. 898–903, 2012.

[Ho20] Hofer, D.; Jäger, M.; Mohamed, A. K. Y. S.; Küng, J.: On Applying Graph Database Time Models for Security Log Analysis. In: FDSE. Vol. 12466. Lecture Notes in Computer Science, Springer, pp. 87–107, 2020.

[Hu06] Huang; Lin; Chen; Fan: Review of outlier detection. Application Research of Computers, Application Research of Computers 8, 2006.

[Kh23] Khalaji, E.; Eraslan, S.; Yesilada, Y.; Yaneva, V.: Effects of data preprocessing on detecting autism in adults using web-based eye-tracking data. Behav. Inf. Technol. 42 (14), pp. 2476–2484, 2023.

[KKM14] Koutras, P.; Katsamanis, A.; Maragos, P.: Predicting Eyes' Fixations in Movie Videos: Visual Saliency Experiments on a New Eye-Tracking Database. In: HCI (23). Vol. 8532. Lecture Notes in Computer Science, Springer, pp. 183–194, 2014.

[Kl16] Klettke, M.; Störl, U.; Shenavai, M.; Scherzinger, S.: NoSQL schema evolution and big data migration at scale. In: IEEE BigData. IEEE Computer Society, pp. 2764–2774, 2016.

[KS19] Kret, M. E.; Sjak-Shie, E. E.: Preprocessing pupil size data: Guidelines and code. Behavior Research Methods 51 (3), pp. 1336–1342, 2019.

[LKK24] Landes, J.; Köppl, S.; Klettke, M.: Data Processing Pipeline for Eye-Tracking Analysis. In: GvDB. Vol. 3710. CEUR Workshop Proceedings, CEUR-WS.org, pp. 35–42, 2024.

[Ne14] Needham, M.: Neo4j: Cypher - Creating a time tree down to the day, Accessed: 2024-08-30, 2014, URL: https://www.markhneedham.com/blog/2014/04/19/neo4j-cypher-creating-a-time-tree-down-to-the-day/.

[Ne24] Neo4j, Inc.: Constraints, Accessed: 2024-05-07, 2024, URL: https://neo4j.com/docs/cypher-manual/current/constraints/.

[No24] Novotný, M.; Fliegel, K.; Simka, M.; Polak, L.; Kufa, J.: Eye-Tracking Data Extension for Omnidirectional Image Quality Assessment Database (OMNIQAD). In: RADIOELEKTRONIKA. IEEE, pp. 1–4, 2024.

[Pi20] Pires, I. M.; Hussain, F.; M. Garcia, N.; Lameski, P.; Zdravevski, E.: Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification. Future Internet 12 (11), Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, p. 194, 2020.

[PN24a] Perini, M.; Nikolic, M.: In-Database Data Imputation. Proc. ACM Manag. Data 2 (1), 2024.

[PN24b] Priyambudi, Z. S.; Nugroho, Y. S.: Which algorithm is better? An implementation of normalization to predict student performance. AIP Conference Proceedings 2926 (1), p. 020110, 2024.

[PVK17] Pokorný, J.; Valenta, M.; Kovacic, J.: Integrity constraints in graph databases. In: ANT/SEIT. Vol. 109. Procedia Computer Science, Elsevier, pp. 975–981, 2017.

[Sc19] Schörgenhumer, A.; Kahlhofer, M.; Chalupar, P.; Grünbacher, P.; Mössenböck, H.: A Framework for Preprocessing Multivariate, Topology-Aware Time Series and Event Data in a Multi-System Environment. In: HASE. IEEE, pp. 115–122, 2019.

[Sc23] Schüle, M. E.; Scalerandi, L.; Kemper, A.; Neumann, T.: Blue Elephants Inspecting Pandas: Inspection and Execution of Machine Learning Pipelines in SQL. In: EDBT. OpenProceedings.org, pp. 40–52, 2023.

[Sh05] Shafranovich, Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files, RFC 4180, 2005, URL: https://www.rfc-editor.org/info/rfc4180.

[SK22] Störl, U.; Klettke, M.: Darwin: A Data Platform for Schema Evolution Management and Data Migration. In: EDBT/ICDT Workshops. Vol. 3135. CEUR Workshop Proceedings, CEUR-WS.org, 2022.

[VPS18] Vinutha, H. P.; Poornima, B.; Sagar, B. M.: Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In (Satapathy, S. C.; Tavares, J. M. R.; Bhateja, V.; Mohanty, J. R., eds.): Information and Decision Sciences. Springer, Singapore, pp. 511–518, 2018.

[ZJZ11] Zhang, S.; Jin, Z.; Zhu, X.: Missing data imputation by utilizing information within incomplete instances. Journal of Systems and Software 84 (3), pp. 452–459, 2011.

[ZT22] Zhang, Y.; Thorburn, P. J.: Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. Future Generation Computer Systems 128, pp. 63–72, 2022.

[ZWS18] Zhao, J.; Wang, W.; Sheng, C.: Data Preprocessing Techniques. In (Zhao, J.; Wang, W.; Sheng, C., eds.): Data-Driven Prediction for Industrial Processes and Their Applications. Springer International Publishing, Cham, pp. 13–52, 2018.

Impact of Preprocessing on Classification Results of Eye-Tracking-Data

Jennifer Landes [1], Meike Klettke [1], and Sonja Köppl [2]

Abstract: Eye-tracking data provides valuable insights into human behavior, but its noisy and unstable nature necessitates robust preprocessing for accurate analysis. This study evaluates a tailored preprocessing pipeline designed to enhance machine learning classifier performance. Unlike prior research focusing on isolated preprocessing steps, this work systematically combines and compares techniques, including missing value imputation, outlier handling, and normalization, specifically optimized for eye-tracking data. The pipeline's impact is tested on classification accuracy, particularly in detecting academic dishonesty. By experimenting with diverse methods for handling missing data, outliers, and feature scaling, we assess their combined effects on classifier performance. A Random Forest classifier is utilized due to its proven effectiveness in prior studies [NS20]. This research not only builds on earlier findings but extends them by optimizing each preprocessing step. Results show a well-designed pipeline significantly enhances classification accuracy, offering insights into optimal preprocessing techniques for behavioral prediction tasks.

Keywords: Data Preprocessing, Random Forest, Classification, Eye-Tracking

1 Introduction

Eye-tracking data has become an invaluable resource for understanding cognitive processes, attentional shifts, and behaviors such as deception in assessments [Bi15]. Despite its potential, the *complexity and noise* inherent in eye-tracking data pose significant challenges [LKK24b]. Individual variability, data fluctuations, and measurement artifacts necessitate advanced preprocessing techniques to extract meaningful signals and ensure robust machine learning outcomes. Without appropriate handling, *noise in raw eye-tracking data* can severely compromise the accuracy and reliability of classification models. So, this paper focuses on the *essential preprocessing steps* required to enhance the accuracy of classification models for eye-tracking datasets. The dataset analyzed originates from an experimental study where students' eye movements were recorded during exam-related tasks to identify predictive markers of cheating behavior—a complex phenomenon not easily detected through traditional methods. Challenges such as high variability, missing values, and outliers in the data necessitate a structured preprocessing approach.

We propose and evaluate a comprehensive preprocessing pipeline addressing key data quality concerns, including managing missing data, normalizing features to reduce individual

[1] University of Regensburg, Data Engineering Group, Regensburg, Germany,
jennifer.landes@stud.uni-regensburg.de, https://orcid.org/0009-0003-1914-598X;
meike.klettke@ur.de, https://orcid.org/0000-0003-0551-8389
[2] Hochschule Neu-Ulm, Business Informatics, Neu-Ulm, Germany,
sonja.koeppl@hnu.de, https://orcid.org/0000-0002-9806-9318

doi:10.18420/BTW2025-127

variability, and employing smoothing techniques to minimize noise. Various strategies for imputation (e.g., mean substitution, Last Observation Carried Forward, and deletion), outlier detection, and feature scaling (e.g., z-score normalization, robust scaling, and min-max scaling) are systematically compared. Using Random Forest classifiers, we assess the impact of these preprocessing steps on classification performance, aiming to identify optimal combinations tailored to the context of classifying and thus analyzing eye-tracking data.

So, the *Research Questions* are:

- **RQ1:** How do different preprocessing methods, such as handling missing values, normalization, and outlier detection, influence the classification accuracy of eye-tracking data?
- **RQ2:** What combination of preprocessing techniques provides the optimal balance between signal clarity and classifier performance for predicting behaviors in eye-tracking datasets?

To address these research questions, we design and evaluate a preprocessing pipeline and analyze its impact on classification accuracy in subsequent sections of the paper. To validate the robustness of the proposed methods, additional experiments are planned, including a second eye-tracking study and analyses of datasets from the cognitive domain. These steps aim to test and refine the preprocessing pipeline across diverse eye-tracking datasets.

Structure. *Chapter 1.1* reviews the prior research done on preprocessing techniques for eye-tracking data and classifier comparisons, providing the foundation for this study. *Chapter 2* presents a literature review, focusing on how preprocessing impacts machine learning performance in eye-tracking data analysis. *Chapter 3* focuses on the design of the preprocessing pipeline, providing the foundation for answering *RQ1*. It describes steps for cleaning, filtering, and normalizing of eye-tracking data as well as the experimental design, data collection process, and the selection of features, followed by implementation of the pipeline and the Random Forest classifier. *Chapter 4* systematically evaluates different preprocessing combinations, addressing RQ2 by identifying the optimal configuration for classification performance by discussing the results how preprocessing combinations influenced accuracy with the Random Forest model. Finally, *Chapter 5* offers an outlook on future research, including further validation and generalization of the preprocessing pipeline to other datasets.

Contribution. This paper makes the following contributions to the field:

1. Identification of needs for the Development of specialized data engineering algorithms for preprocessing eye-tracking data.
2. Design of a comprehensive framework for composing and evaluating data engineering pipelines tailored to eye-tracking datasets.
3. Introduction of a methodology to systematically generate, evaluate, and optimize data engineering pipelines to improve preprocessing outcomes.

1.1 Contextualization of the Work and Preliminary Research

The analysis of eye-tracking data unfolds in multiple phases: literature review, experimental design, data collection, preprocessing pipeline design, and the application of machine learning techniques for classification. Initially, *a quantitative survey* gathers insights into students´ cheating behaviors and contextual factors influencing them [LKK24b]. The results of this survey formed the basis for conceptualizing and designing an eye-tracking experiment to detect cheating during online exams, aligning with the project ii.oo - digital kompetenzorientiert Prüfen. The study aims to explore factors and methods influencing academic misconduct, a timely issue exacerbated by the shift to online exams during COVID-19 [Ja21]. Prior research addressed *preprocessing pipeline design* for eye-tracking data and *evaluating machine learning classifiers* to identify the most effective model for predicting cheating behavior. Our approach introduces a modular framework for systematically testing preprocessing steps and their combined effects on classifier performance. Our prior article showed Random Forest outperformed Support Vector Machines, k-Nearest Neighbors, and Decision Trees, accurately classifying participants into cheating and non-cheating groups. Building on these findings, this work applies the optimized preprocessing pipeline and Random Forest classifier for further analysis [LKK24b, LKK24a].

2 Related Work

Related articles emphasize the critical role of preprocessing in optimizing machine learning models. Systematic evaluation of preprocessing steps can lead to significant accuracy improvements. This underscores the need for domain-specific preprocessing guidelines to enhance model performance and generalizability. The importance of data preprocessing in machine learning has grown with the increasing complexity and scale of datasets [Mi95]. Prior studies, such as those by Burdack et al. (2020) and Zheng et al. (2018) have demonstrated how optimized preprocessing steps enhance model performance. This study builds on these insights, comparing preprocessing techniques for eye-tracking data.

Preprocessing directly affects the performance of machine learning models, as highlighted in diverse domains. For instance, [Bu20] showed that in gait classification, steps like filtering and data reduction significantly improve classifiers like SVMs and CNNs, whereas normalization offers limited benefits. Similarly, Kamiran and Calders [KC12] explored preprocessing to mitigate discrimination in classification, demonstrating that strategies like resampling and reweighting can enhance fairness without compromising accuracy. Furthermore, [Vo21] introduced a novel approach by converting time series into two-dimensional images for CNN-based classification, significantly improving accuracy over traditional statistical methods. In industrial applications, Zheng et al. [ZWOM18] highlighted that multichannel preprocessing, treating data axes as overlapping color channels, enhances activity recognition models. Similarly, [Mi20] demonstrated the advantages of ensemble preprocessing techniques in chemometrics, improving robustness and applicability across

datasets. Environmental applications also underscore the importance of preprocessing. For example, [Gh15] evaluated methods like derivative preprocessing in predicting soil heavy metal concentrations, finding that tailored preprocessing enhances accuracy.

3 Design and Evaluation of Preprocessing Pipeline

This chapter addresses *RQ1* by systematically exploring preprocessing methods such as missing value imputation, normalization, and outlier detection, smoothing and filtering evaluating their impact on classification performance. Variations and combinations of these steps create a comprehensive search space. The study uses eye-tracking data from an experiment on detecting cheating in digital exams. Implementation in Python evaluates preprocessing strategies, with classification accuracy as the benchmark.

3.1 Design of the Preprocessing Pipeline

The preprocessing pipeline methodically prepares eye-tracking data for analysis by progressively refining the dataset to ensure accurate and reliable results. The aim is to maximize the accuracy of the classification of the eye-trackig data. The series of steps is designed to first remove errors and noise, followed by detailed refinement to guarantee that the examination of eye-tracking patterns is precise. The preprocessing pipeline $\mathcal{P}$ is defined as a sequence of transformations applied to the raw dataset D, producing a processed dataset D': $\mathcal{P} : D \rightarrow D'$. Each feature $f_i \in F = \{f_1, f_2, \ldots, f_m\}$ undergoes a specific sequence of preprocessing steps $\mathcal{P}_i(f_i)$, tailored to its characteristics. For each feature f, the pipeline is described as:

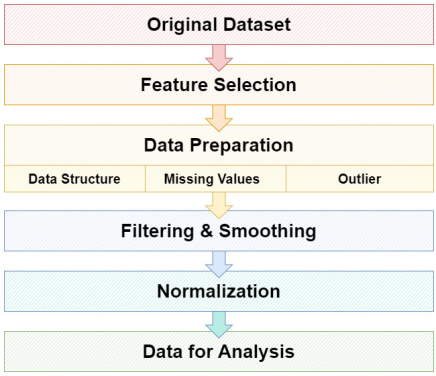


Fig. 1: Preprocessing Pipeline for Eye-Tracking Data

$$\mathcal{P}(f) = P_{\text{limits}} \circ P_{\text{filter}} \circ P_{\text{smooth}} \circ P_{\text{norm}} \circ P_{\text{out}} \circ P_{\text{mv}}(f)$$

Where each part P of the pipeline represents the corresponding preprocessing step: P_{limits} for applying feature limits, P_{filter} for applying Butterworth Filter, P_{smooth} for applying Smoothing, P_{norm} for Normalization, P_{out} for Outlier Detection and Handling, P_{mv} for Missing Value Detection and Handling. The preprocessing pipeline is systematically evaluated over all combinations (search space) of methods. Let $\mathcal{P}_j$ represent a specific pipeline configuration. The performance $\mathcal{A}$ of $\mathcal{P}_j$ is evaluated using classification accuracy: $\mathcal{A}(\mathcal{P}_j) = \text{Accuracy}(\text{Classifier}(\mathcal{P}_j(F)))$. So, the pipeline $\mathcal{P}^*$ is determined by maximizing

accuracy: $\mathcal{P}^* = \max_{\mathcal{P}_j} \mathcal{A}(\mathcal{P}_j)$. The preprocessing pipeline provides a modular design for the characteristics of eye-tracking data. The preprocessing pipeline is applied to *selected features*, which are relevant for the analysis. Each feature may require a special combination of methods. These comprise Gaze X/Y, Pupil Size, Gaze velocity, Gaze Acceleration, Fixation X/Y, Fixation Dispersion, Saccade Amplitude, Saccade Peak Velocity,Saccade Peak Acceleration, Saccade Peak Deceleration, Saccade Direction, and Saccade X/Y.

This feature selection is followed by Data Preparation, which includes Data Structure, and Detection and Handling of Missing Values and Outliers. *Data Structure* involves validation of rows, eliminating obvious errors, and inconsistencies within the dataset and number format. Then, *missing values* are detected and handled with imputation or deletion, which depends on each feature [ZT22, NUS21]. Following this, *Outlier Detection* is performed to identify atypical eye movement data points early in the process. Correcting these outliers is vital, as they can significantly skew analysis outcomes [VPS18]. In the next phase, *Normalization* is conducted to scale the data to a uniform range. This standardization ensures that all features are balanced [Pi20, PN24, PS15, KSS19].

The next phase involves optionally filtering and smoothing techniques for the special nature of Eye-Traxking data (e.g. noise). One method is to apply a *Low-Pass Filter* to the data, which serves to reduce high-frequency noise and rapid fluctuations present in the eye-tracking measurements [AH10]. This filtering process smooths out erratic variations, thereby enhancing the overall quality and consistency of the data [Ol07, Ra23]. Another filter technique is done by setting a maximum and minimum *threshold* for selected features. To further improve data quality, *Data Smoothing* by use of the Fourier-Transformation is employed to reduce minor fluctuations that may obscure important underlying patterns in the eye-tracking data. This step amplifies the clarity of significant patterns, facilitating more accurate analysis [LKK24b, Li21].

3.2 Extension of the Preprocessing Pipeline

To enhance the accuracy of classification, it is planned to systematically test and evaluate each step of the preprocessing pipeline. This involves experimenting with various methods for the preprocessing steps and evaluating their impact on classification performance. The variations of each preprocessing step are displayed in Figure 2.

The first step in the pipeline involves the treatment of *missing values and outliers*, as these can significantly skew results if not properly managed. The methods to be explored are shown in Table 1.

Handling missing values ($P_{\mathbf{mv}}$) is defined as:

$$P_{\mathrm{mv}}(f) = \begin{cases} \mathrm{mean}(f), & \text{imputation with mean} \\ \mathrm{locf}(f), & \text{imputation with Last Observation Carried Forward} \\ f \setminus f_{\mathrm{missing}}, & \text{deletion of missing values} \end{cases}$$

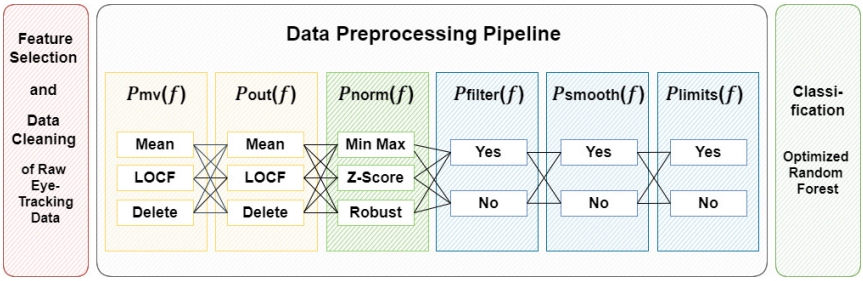


Fig. 2: Search Space of Preprocessing Methods

Outlier Detection and Handling (P_{out}):

Outliers (P_{out}) are identified using z-score thresholding: $f_{\text{out}} = \{x \in f \mid |z(x)| > z_{\text{threshold}}\}$, $z(x) = \frac{x-\mu}{\sigma}$, where μ is the mean and σ is the standard deviation. Outlier handling is defined as:

$$P_{\text{out}}(f) = \begin{cases} \text{mean}(f_{\text{out}}), & \text{imputation with mean} \\ \text{locf}(f_{\text{out}}), & \text{imputation with Last Observation Carried Forward} \\ f \setminus f_{\text{out}}, & \text{delete outliers} \end{cases}$$

Tab. 1: Methods for Handling Missing Values and Outliers [VPS18, ZT22, JDP22]

Imputation Method	**Description**
Mean	Replace missing values with the mean of the corresponding feature to maintain continuity.
Last Observation Carried Forward (LOCF)	Impute missing values by carrying forward the last observed value, useful in time-series data.
Deletion	Remove rows or columns with missing values, especially when the proportion of missing data is small.

Next, *feature scaling and normalization* are focussed, essential for ensuring that all features contribute equally to the classification model. Table 2 summarizes the scaling methods under consideration.

Normalization methods (P_{norm}) include:

$$P_{\text{norm}}(f) = \begin{cases} \frac{f - f_{\min}}{f_{\max} - f_{\min}}, & \text{Min-Max Scaling} \\ \frac{f-\mu}{\sigma}, & \text{Z-score Standardization} \\ \frac{f - \text{median}(f)}{\text{IQR}(f)}, & \text{Robust Scaling} \end{cases}$$

Further refinement of the data involves *setting thresholds, applying low-pass filters, and smoothing* [LKK24b]. Table 3 details these techniques.

Tab. 2: Feature Scaling and Normalization Methods [PN24, Pi20]

Normalization Method	Description
Min-Max Scaling	Scales data to a fixed range (usually [0, 1]).
Z-Score Standardization	Standardizes features by removing the mean and scaling to unit variance, where μ is the mean and σ the standard deviation.
Robust Scaler	Scales data according to the interquartile range, reducing sensitivity to outliers, where IQR is the interquartile range.

Smoothing (P_{smooth}) can be applied using:

$$P_{\text{smooth}}(f) = \begin{cases} \mathcal{F}^{-1}(\mathcal{F}(f)), & \text{apply Fourier Transformation} \\ f, & \text{no smoothing} \end{cases}$$

Low-pass filtering (P_{filter}) uses a Butterworth filter:

$$P_{\text{filter}}(f) = \begin{cases} \text{Butter}(f, \text{cutoff}, \text{order}), & \text{apply Butterworth filter} \\ f, & \text{no filter} \end{cases}$$

Feature limits (P_{limits}) ensure data validity:

$$P_{\text{limits}}(f) = \begin{cases} f, & \text{if } f \in [l_{\min}, l_{\max}] \\ \text{NaN}, & \text{if } f \notin [l_{\min}, l_{\max}] \end{cases}$$

Tab. 3: Data Refinement Techniques [Li21, Ol07]

Data Refinement Technique	Description
Setting Thresholds on Features	Defines upper and lower limits for each feature to exclude extreme values, where $l_{\min}$ and $l_{\max}$ are bounds.
Low-Pass Filtering	Reduces high-frequency noise, smoothing out rapid fluctuations, where cutoff is cutoff frequency and order is the Filter steepness.
Fourier Transformation for Smoothing	Further smooths data by filtering out high-frequency components, emphasizing underlying eye movement patterns. Fourier transform equation is: $\mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\nu t}dt$

3.3 Experiment and Data Set

The experiment collected eye-tracking data from 25 students at Hochschule Neu-Ulm completing five types of exam tasks while cheating or not cheating. So, each dataset was split into five parts, this resulted in total in 125 csv sheets. The students labelled for each task, if they did cheat or not. Sensor data exported via iMotions 5.1 in CSV format was standardized, with columns aligned to resolve discrepancies across devices. Non-essential information, such as PC CPU values, StimType, and Eventsource, was excluded. Table 4 highlights the selected features crucial for analyzing gaze behavior, while system-level metrics and metadata were omitted as irrelevant to the classification task.

Features	Description
GazeLeft and GazeRight	The X and Y coordinates of the left and right eye.
PupilLeft, PupilRight	The pupil size of the left and right eye.
DistanceLeft, DistanceRight	The distance of the left and right eye.
CameraLeft, CameraRight	The camera coordinates of the left and right eye.
ValidityLeft, ValidityRight	The validity of the gaze data from the left and right eye.
Gaze X, Gaze Y	The X and Y coordinates of the participant's gaze point.
Interpolated Gaze, Distance	Interpolated X and Y coordinates and distance of the gaze.
Gaze Velocity, Gaze Acceleration	The velocity and acceleration of the gaze point.
Saccade X, Saccade Y	Rapid eye movements between fixations.
Saccade Duration	The time it takes to move from one point to another.
Fixation Duration	Time interval from beginning to the end of a fixation.
Fixation X, Fixation Y	The X and Y coordinates of the participant's fixation point.

Tab. 4: Features of Eye-Tracking Dataset.

3.4 Implementation

The implementation systematically tests combinations of preprocessing methods to address *RQ1* by analyzing individual impacts and *RQ2* by identifying the optimal pipeline. Using gaze-tracking data, each combination's effect on model accuracy is evaluated with a `RandomForestClassifier`, and results are stored for analysis. Future work will explore additional features to improve the model. The implementation uses `pandas`, `numpy`, and `scikit-learn` for preprocessing and modeling. Signal processing operations, such as applying filters, are performed using the `scipy.signal` module. The data is loaded from CSV, handling encodings and converting all data to numeric form, replacing non-numeric values with `NaN`. To address *missing data*, imputation by mean or LOCF or deletion is done.

```
def handle_missing_values(df, method):
    if method == 'mean':
        df.fillna(df.mean(), inplace=True)
    elif method == 'locf':
        df.fillna(method='ffill', inplace=True)
    elif method == 'delete':
        df.dropna(inplace=True)
    return df
```

List. 1: Python function to handle missing values

Next, *outlier detection* is implemented using a z-score threshold of 3. Identified outliers are managed using one of three approaches: replacing them with the mean, LOCF, or deleting the rows. This step ensures the dataset's integrity by reducing distortions caused by extreme values. The Python function iterates over each column to calculate z-scores, identifies values exceeding the threshold, and applies the chosen method to handle these outliers.

```
def handle_outliers(df, method, z_thresh=3):
    z_scores = np.abs((df - df.mean()) / df.std())
```

```
    outliers = (z_scores > z_thresh)
    if method == 'mean':
        for col in df.columns:
            df.loc[outliers[col], col] = df[col].mean()
    elif method == 'locf':
        for col in df.columns:
            df.loc[outliers[col], col] = df[col].ffill()
    elif method == 'delete':
        df = df[~outliers.any(axis=1)]
    return df
```

List. 2: Python function to handle outliers

Data normalization is carried out by MinMaxScaler, which scales features to a specified range; RobustScaler, which reduces the influence of outliers by scaling the data based on percentiles; or StandardScaler, which standardizes the data by centering it and scaling it to unit variance (z-score normalization). The normalization step is implemented as follows:

```
def normalize_data(df, method):
    if method == 'minmax':
        scaler = MinMaxScaler()
    elif method == 'robust':
        scaler = RobustScaler()
    elif method == 'zscore':
        scaler = StandardScaler()
    df[df.columns] = scaler.fit_transform(df[df.columns])
    return df
```

List. 3: Python function to normalize data

The last three steps are optional. First, the application of *feature limits*. This ensures that the values of the features `'Gaze X'` and `'Gaze Y'` lie within realistic bounds (dimensions of a screen, from 0 to 1920 for X and 0 to 1080 for Y), filtering out noisy data points. Second, for *data smoothing*, the function applies a Fourier transformation to reduce noise, which stabilizes the data. Third, the pipeline includes the option to apply a *low-pass filter*. This function filters out high-frequency noise from the data using a Butterworth filter, which passes signals below a cutoff frequency and attenuates higher frequencies.

```
def apply_feature_limits(df, limits=True):
    if limits:
        df = df[(df['Gaze X'] >= 0) & (df['Gaze X'] <= 1920)]
        df = df[(df['Gaze Y'] >= 0) & (df['Gaze Y'] <= 1080)]
    return df
```

List. 4: Python function to apply limits

Once preprocessing is complete, a `target` label is added to indicate whether the data represents a 'cheat' or 'no cheat' scenario (1 for cheating and 0 for not cheating). The process

is handled by the `load_and_preprocess_data` function, which iterates through the CSV files and applies preprocessing steps. The function systematically tests various *combinations of preprocessing steps* by generating all possible configurations. Each combination is evaluated by feeding the preprocessed data into the pipeline. The data is split into training and testing sets (80-20 split), and a `RandomForestClassifier` is trained on the training data. After the model is trained, its accuracy is evaluated on the test set using the `accuracy_score` metric.

```
clf = RandomForestClassifier(max_depth=None, max_features='log2',
                             min_samples_leaf=2, min_samples_split=2,
                             n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
accuracy = accuracy_score(y_test, clf.predict(X_test))
```

List. 5: Python function for classification

This modular and systematic approach enables efficient exploration of preprocessing strategies, ensuring robust performance analysis and paving the way for future research enhancements. The full implementation code is available in the appendix.

3.5 Results and Discussion

The results presented in this section address *RQ1* by evaluating the effects of individual preprocessing methods and *RQ2* by comparing the performance of various preprocessing configurations. The implementation calculated the accuracy for each combination and this resulted in a list of 217 results. The *highest accuracy* values were observed in combinations involving *LOCF for missing values, Z-score normalization*, and *LOCF* or *mean for outlier handling*. These findings emphasize the importance of tailoring preprocessing pipelines to the specific challenges of eye-tracking data. Unlike previous studies that often used heuristic or general-purpose methods, this procedure systematically observes preprocessing techniques to improve classification accuracy. Accuracies in this study reached up to 0.920, with the highest observed accuracy achieved using LOCF for both missing values and outliers, Z-score normalization, feature limits, smoothing, and no filtering. Table 5 summarizes the results of the tested combinations on the eye-tracking dataset.

The results show further, that *Z-score normalization* outperformed Min-Max and robust normalization. By standardizing data through mean and scaling to unit variance, Z-score normalization aligns well with Random Forest, as observed in gait analysis [Bu20]. It led to an average 3.5% improvement over Min-Max scaling. LOCF for outlier handling consistently yielded high accuracy, especially with Z-score normalization and smoothing. Mean-based outlier handling also performed well when paired with LOCF for missing values and Z-score normalization. Applying *feature limits* slightly improved accuracy compared to when this option was false. Similarly, *smoothing improved model performance*, likely by reducing high-frequency noise while preserving patterns. Interestingly, no filtering

MV	Outlier	Normalization	Limits	Smoothing	Filtering	Accuracy
locf	locf	zscore	True	True	False	0.920491232
locf	locf	minmax	True	True	False	0.919774686
locf	mean	robust	True	True	False	0.919587588
locf	mean	zscore	True	True	False	0.919531856
locf	mean	minmax	True	True	False	0.919328835
locf	locf	robust	True	True	False	0.919209411
locf	delete	zscore	True	True	False	0.919089089
locf	delete	minmax	True	True	False	0.918783522
locf	delete	robust	True	True	False	0.918369398
mean	delete	minmax	True	True	False	0.915529501

Tab. 5: Highest Values: Combinations of Preprocessing Methods and Corresponding Accuracy.

often outperformed filtering, suggesting filtering techniques might have removed valuable information or distorted the data. These findings highlight the need to align preprocessing methods with dataset characteristics.

MV	Outlier	Normalization	Limits	Smoothing	Filter	Accuracy
delete	delete	minmax	True	False	False	0.732716097
delete	delete	minmax	False	False	False	0.732716097
mean	locf	robust	True	False	True	0.732228157
mean	locf	robust	False	False	True	0.732228157
mean	mean	robust	True	False	True	0.730537779
mean	mean	robust	False	False	True	0.730537779
mean	locf	zscore	True	False	True	0.725283685
mean	locf	zscore	False	False	True	0.725283685
mean	locf	zscore	True	False	False	0.720188686
mean	locf	zscore	False	False	False	0.720188686

Tab. 6: Lowest Values: Comparison of Preprocessing Options and their Impact on Accuracy.

On the other side, Table 6 lists the 10 lowest accuracies from preprocessing methods. Mean imputation combined with robust normalization generally yielded *lower accuracies*, especially with smoothing, reflecting trends in gait analysis [Bu20]. The lowest accuracy (0.7202) occurred with *mean imputation, LOCF for outlier handling, and Z-score normalization, without limits or filtering*. Mean imputation underperformed as it failed to retain temporal patterns, while robust normalization was less effective than Z-score normalization. Absence of smoothing amplified noise, and poor filtering either removed key signals or allowed noise. The low-performing combinations underscore the importance of temporal dynamics, with LOCF preserving continuity better than mean imputation. Z-score normalization offered superior noise reduction without over-filtering, enhancing signal quality.

4 Conclusion

This study explores the impact of preprocessing techniques on the classification accuracy of eye-tracking data, addressing challenges like noise, variability, and missing values. By systematically evaluating combinations of preprocessing methods, we identified approaches that significantly improve the accuracy of machine learning classifiers, particularly for detecting behavioral patterns such as academic dishonesty. By designing and evaluating a preprocessing pipeline, this research highlights the role of data preparation in improving classification accuracy of Eye-tracking data. This research contributes a robust framework for preprocessing eye-tracking data, providing actionable insights into designing tailored pipelines for behavioral analysis.

Regarding *RQ1*, our findings demonstrate that preprocessing methods play a critical role in enhancing classifier performance. Techniques like Last Observation Carried Forward (LOCF) for missing values and outliers, combined with Z-score normalization, consistently yielded higher accuracy, underscoring the importance of preserving temporal dynamics and standardizing features. In case of *RQ2*, the optimal pipeline configuration included LOCF for both missing values and outliers, Z-score normalization, feature limits, smoothing, and no filtering. This combination achieved the highest accuracy, effectively balancing noise reduction and the preservation of meaningful patterns. Furthermore, smoothing techniques, and noise reduction were critical in enhancing signal stability and reducing high-frequency noise, though care is needed to avoid excessive filtering.

5 Outlook on Future Research

Future research will extend the optimized preprocessing pipeline to diverse datasets. A planned follow-up eye-tracking study under controlled conditions will validate its adaptability and assess its performance with higher-quality data. Additionally, the pipeline's applicability to other physiological data, such wearable sensors, will be explored, addressing shared challenges like noise, variability, and missing values, while testing its scalability across other datasets. A time series approach will be incorporated to better capture temporal dynamics in eye-tracking data, enabling the detection of sequential patterns and trends that inform preprocessing and classification strategies. Improving data quality and conducting comprehensive data profiling will also be a priority, uncovering patterns, anomalies, and distributions that inform optimal preprocessing strategies. These efforts will refine each pipeline step, ensuring they enhance the reliability and interpretability of classification models. Advanced imputation techniques, adaptive filtering methods, and hybrid normalization strategies will be systematically evaluated to further improve the pipeline's generalizability and classification accuracy. By bridging theoretical preprocessing strategies and practical applications, this research aims to create a robust, universal preprocessing framework. Expanding its capabilities to broader datasets could transform data analysis in cognitive research, education, and healthcare, enabling impactful applications in behavioral and physiological data analysis.

Bibliography

[AH10] Adali, Tulay; Haykin, Simon: Adaptive Signal Processing: Next Generation Solutions. John Wiley & Sons, June 2010. Google-Books-ID: RaS5h_dabagC.

[Bi15] Bixler, Robert; Blanchard, Nathaniel; Garrison, Luke; D'Mello, Sidney: Automatic Detection of Mind Wandering During Reading Using Gaze and Physiology. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, Seattle Washington USA, pp. 299–306, November 2015.

[Bu20] Burdack, Johannes; Horst, Fabian; Giesselbach, Sven; Hassan, Ibrahim; Daffner, Sabrina; Schöllhorn, Wolfgang I.: Systematic Comparison of the Influence of Different Data Preprocessing Methods on the Performance of Gait Classifications Using Machine Learning. Frontiers in Bioengineering and Biotechnology, 8, 2020.

[Gh15] Gholizadeh, Asa; Boruvka, Lubos; Saberioon, Mehdi; Kozák, Josef; Vašát, Radim; Nemecek, Karel: Comparing Different Data Preprocessing Methods for Monitoring Soil Heavy Metals Based on Soil Spectral Features. Soil and Water Research, 10:218–227, November 2015.

[Ja21] Janke, Stefan; Rudert, Selma C.; Änne Petersen; Fritz, Tanja M.; Daumiller, Martin: Cheating in the wake of COVID-19: How dangerous is ad-hoc online testing for academic integrity? Computers and Education Open, 2:100055, 2021.

[JDP22] Joel, Luke Oluwaseye; Doorsamy, Wesley; Paul, Babu Sena: A review of missing data handling techniques for machine learning. International Journal of Innovative Technology and Interdisciplinary Sciences, 5(3):971–1005, 2022.

[KC12] Kamiran, Faisal; Calders, Toon: Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems, 33(1):1–33, October 2012.

[KSS19] Kret, Mariska E.; Sjak-Shie, Elio E.: Preprocessing pupil size data: Guidelines and code. Behavior Research Methods, 51(3):1336–1342, June 2019.

[Li21] Li, Yuanfeng; Deng, Jiangang; Wu, Qun; Wang, Ying: Eye-Tracking Signals Based Affective Classification Employing Deep Gradient Convolutional Neural Networks. December 2021. Accepted: 2022-05-09T10:43:19Z Publisher: International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI).

[LKK24a] Landes, Jennifer; Köppl, Sonja; Klettke, Meike: Comparison of Classifiers for Eye-Tracking Data. In (Klein, Maike; Krupka, Daniel; Winter, Cornelia; Gergeleit, Martin; Martin, Ludger, eds): 54. Jahrestagung der Gesellschaft für Informatik, INFORMATIK 2024 - Lock in or log out? Wie digitale Souveränität gelingt, Wiesbaden, Germany, September 24-26, 2024. volume P-352 of LNI. Gesellschaft für Informatik, Bonn, pp. 1449–1462, 2024.

[LKK24b] Landes, Jennifer; Köppl, Sonja; Klettke, Meike: Data Processing Pipeline for Eye-Tracking Analysis. In (Störl, Uta, ed.): Proceedings of the 35th GI-Workshop Grundlagen von Datenbanken, Herdecke, Germany, May 22-24, 2024. volume 3710 of CEUR Workshop Proceedings. CEUR-WS.org, pp. 35–42, 2024.

[Mi95] Mitchell, R. Scott: The application of machine learning techniques to time-series data, 1995.

[Mi20] Mishra, Puneet; Biancolillo, Alessandra; Roger, Jean Michel; Marini, Federico; Rutledge, Douglas N.: New data preprocessing trends based on ensemble of multiple preprocessing techniques. TrAC Trends in Analytical Chemistry, 132:116045, November 2020.

[NS20] Nurwulan, Nurul Retno; Selamaj, Gjergji: Random Forest for Human Daily Activity Recognition. Journal of Physics: Conference Series, 1655(1):012087, October 2020.

[NUS21] Nugroho, Heru; Utama, Nugraha Priya; Surendro, Kridanto: Normalization and outlier removal in class center-based firefly algorithm for missing value imputation. Journal of Big Data, 8(1):129, October 2021.

[Ol07] Olsson, Pontus: Real-time and Offline Filters for Eye Tracking. 2007.

[Pi20] Pires, Ivan Miguel; Hussain, Faisal; M. Garcia, Nuno; Lameski, Petre; Zdravevski, Eftim: Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification. Future Internet, 12(11):194, November 2020. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[PN24] Priyambudi, Zulfikar Setyo; Nugroho, Yusuf Sulistyo: Which algorithm is better? An implementation of normalization to predict student performance. AIP Conference Proceedings, 2926(1):020110, January 2024.

[PS15] Patro, S. Gopal Krishna; Sahu, Kishore Kumar: Normalization: A Preprocessing Stage, March 2015. arXiv:1503.06462 [cs].

[Ra23] Raju, Mehedi H.; Friedman, Lee; Bouman, Troy M.; Komogortsev, Oleg V.: Filtering Eye-Tracking Data From an EyeLink 1000: Comparing Heuristic, Savitzky-Golay, IIR and FIR Digital Filters. Journal of Eye Movement Research, 14(3), October 2023. arXiv:2303.02134 [cs].

[Vo21] Vortmann, Lisa-Marie; Knychalla, Jannes; Annerer-Walcher, Sonja; Benedek, Mathias; Putze, Felix: Imaging Time Series of Eye Tracking Data to Classify Attentional States. Frontiers in Neuroscience, 15, May 2021. Publisher: Frontiers.

[VPS18] Vinutha, H. P.; Poornima, B.; Sagar, B. M.: Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In (Satapathy, Suresh Chandra; Tavares, Joao Manuel R.S.; Bhateja, Vikrant; Mohanty, J. R., eds): Information and Decision Sciences. Springer, Singapore, pp. 511–518, 2018.

[ZT22] Zhang, Yifan; Thorburn, Peter J.: Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. Future Generation Computer Systems, 128:63–72, March 2022.

[ZWOM18] Zheng, Xiaochen; Wang, Meiqing; Ordieres-Meré, Joaquín: Comparison of Data Preprocessing Approaches for Applying Deep Learning to Human Activity Recognition in the Context of Industry 4.0. Sensors, 18(7):2146, July 2018. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

A Appendix I: Python Code

Attached are the methods implemented in Python for the Preprocessing steps mentioned above.

Handling of for Missing Values.

```
# Function to handle missing values
def handle_missing_values(df, method):
    if method == 'mean':
        df.fillna(df.mean(), inplace=True)
    elif method == 'locf':
        df.fillna(method='ffill', inplace=True)
    elif method == 'delete':
        df.dropna(inplace=True)
    return df
```

Outlier Detection and Handling.

```
# Function to handle outliers
def handle_outliers(df, method, z_thresh=3):
    z_scores = np.abs((df - df.mean()) / df.std())
    outliers = (z_scores > z_thresh)
    if method == 'mean':
        for col in df.columns:
            df.loc[outliers[col], col] = df[col].mean()
    elif method == 'locf':
        for col in df.columns:
            df.loc[outliers[col], col] = df[col].ffill()
    elif method == 'delete':
        df = df[~outliers.any(axis=1)]
    return df
```

Data Normalization.

```
# Normalization function
def normalize_data(df, method):
    if method == 'minmax':
        scaler = MinMaxScaler()
    elif method == 'robust':
        scaler = RobustScaler()
    elif method == 'zscore':
        scaler = StandardScaler()
    df[df.columns] = scaler.fit_transform(df[df.columns])
    return df
```

Optional Filter Methods.

```
# Function to apply feature limits
def apply_feature_limits(df, limits=True):
    if limits:
        df = df[(df['Gaze X'] >= 0) & (df['Gaze X'] <= 1920)]
        df = df[(df['Gaze Y'] >= 0) & (df['Gaze Y'] <= 1080)]
    return df

# Function to smooth data using Fourier transformation
def smooth_data(df, apply_smoothing=True):
    if apply_smoothing:
        for column in df.columns:
            df[column] = np.fft.ifft(np.fft.fft(df[column]).real).real
    return df
# Function to apply a low-pass filter
def apply_low_pass_filter(df, apply_filter=True, cutoff=0.1, fs=30.0):
    if apply_filter:
        nyq = 0.5 * fs
        normal_cutoff = cutoff / nyq
        b, a = butter(1, normal_cutoff, btype='low', analog=False)
        df[df.columns] = filtfilt(b, a, df[df.columns], axis=0)
    return df
```

Method for Preprocessing and Classification.

```
# Function to load and preprocess CSV files
def load_and_preprocess_data(folder_cheat, folder_no_cheat, missing_method,
↪  outlier_method, normalize_method, feature_limits, apply_smoothing,
↪  apply_filter):

# Combinations of preprocessing steps
missing_methods = ['mean', 'locf', 'delete']
outlier_methods = ['mean', 'locf', 'delete']
normalize_methods = ['minmax', 'robust', 'zscore']
feature_limits_options = [True, False]
smoothing_options = [True, False]
filter_options = [True, False]

combinations = list(itertools.product(missing_methods, outlier_methods,
↪  normalize_methods, feature_limits_options, smoothing_options,
↪  filter_options))

# Iterate over all combinations
for combo in combinations:
    missing_method, outlier_method, normalize_method, feature_limits,
    ↪  apply_smoothing, apply_filter = combo
```

```
    data = load_and_preprocess_data(folder_cheat, folder_no_cheat,
    ↪ missing_method, outlier_method, normalize_method, feature_limits,
    ↪ apply_smoothing, apply_filter)
    if not data.empty and not data.isnull().values.any():
        X = data.drop('target', axis=1)
        y = data['target']
        X_train, X_test, y_train, y_test = train_test_split(X, y,
        ↪ test_size=0.2, random_state=42)
        clf = RandomForestClassifier(max_depth=None, max_features='log2',
        ↪ min_samples_leaf=2, min_samples_split=2, n_estimators=100,
        ↪ random_state=42)
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
```

B Appendix II: Accuracies

MV	Outlier	Normalization	Limits	Smoothing	Filter	Accuracy
locf	locf	robust	True	True	False	0,919209
locf	locf	zscore	True	True	False	0,920491
locf	delete	minmax	True	True	False	0,918784
locf	delete	robust	True	True	False	0,918369
locf	delete	zscore	True	True	False	0,919089
mean	mean	minmax	True	True	True	0,870215
mean	mean	minmax	True	True	False	0,915191
mean	mean	minmax	True	False	True	0,744188
mean	mean	minmax	True	False	False	0,863119
mean	mean	minmax	False	True	True	0,870215
mean	mean	minmax	False	True	False	0,915191
mean	mean	minmax	False	False	True	0,744188
mean	mean	minmax	False	False	False	0,863119
mean	mean	robust	True	True	True	0,870962
mean	mean	robust	True	True	False	0,857539
mean	mean	robust	True	False	False	0,753507
mean	mean	robust	False	True	True	0,870962
mean	mean	robust	False	True	False	0,857539
mean	mean	robust	False	False	False	0,753507
mean	mean	zscore	True	True	True	0,869177
mean	mean	zscore	True	True	False	0,915489
mean	mean	zscore	True	False	True	0,733716
mean	mean	zscore	True	False	False	0,862519
mean	mean	zscore	False	True	True	0,869177
mean	mean	zscore	False	True	False	0,915489
mean	mean	zscore	False	False	True	0,733716
mean	mean	zscore	False	False	False	0,862519
mean	locf	minmax	True	True	True	0,874121
mean	locf	minmax	True	True	False	0,912502
mean	locf	minmax	True	False	True	0,75725
mean	locf	minmax	True	False	False	0,858219
mean	locf	minmax	False	True	True	0,874121
mean	locf	minmax	False	True	False	0,912502
mean	locf	minmax	False	False	True	0,75725
mean	locf	minmax	False	False	False	0,858219
mean	locf	robust	True	True	True	0,872016
mean	locf	robust	True	True	False	0,859074
mean	locf	robust	True	False	False	0,75091

Tab. 7: Comparison of Preprocessing Options and their Impact on Accuracy.

MV	Outlier	Normalization	Limits	Smoothing	Filter	Accuracy
mean	locf	robust	False	True	True	0,872016
mean	locf	robust	False	True	False	0,859074
mean	locf	robust	False	False	False	0,75091
mean	locf	zscore	True	True	True	0,869797
mean	locf	zscore	True	True	False	0,835182
mean	locf	zscore	False	True	True	0,869797
mean	locf	zscore	False	True	False	0,835182
mean	delete	minmax	True	True	True	0,871903
mean	delete	minmax	True	False	True	0,74736
mean	delete	minmax	True	False	False	0,860461
mean	delete	minmax	False	True	True	0,871903
mean	delete	minmax	False	False	True	0,74736
mean	delete	minmax	False	False	False	0,860461
mean	delete	robust	True	True	True	0,872625
mean	delete	robust	True	True	False	0,861512
mean	delete	robust	True	False	True	0,733661
mean	delete	robust	True	False	False	0,751847
mean	delete	robust	False	True	True	0,872625
mean	delete	robust	False	True	False	0,861512
mean	delete	robust	False	False	True	0,733661
mean	delete	robust	False	False	False	0,751847
mean	delete	zscore	True	True	True	0,870248
mean	delete	zscore	True	True	False	0,914647
mean	delete	zscore	True	False	True	0,736469
mean	delete	zscore	True	False	False	0,861426
mean	delete	zscore	False	True	True	0,870248
mean	delete	zscore	False	True	False	0,914647
mean	delete	zscore	False	False	True	0,736469
mean	delete	zscore	False	False	False	0,861426
locf	mean	minmax	True	True	True	0,875337
locf	mean	minmax	True	False	True	0,749921
locf	mean	minmax	True	False	False	0,872081
locf	mean	robust	True	True	True	0,879564
locf	mean	robust	True	False	True	0,752226
locf	mean	robust	True	False	False	0,873382
locf	mean	zscore	True	True	True	0,875827
locf	mean	zscore	True	False	True	0,750097
locf	mean	zscore	True	False	False	0,877291
locf	locf	minmax	True	True	True	0,876698
locf	locf	minmax	True	False	True	0,750773
locf	locf	minmax	True	False	False	0,87058
locf	locf	robust	True	True	True	0,88001
locf	locf	robust	True	False	True	0,752425
locf	locf	robust	True	False	False	0,871774
locf	locf	zscore	True	True	True	0,877952
locf	locf	zscore	True	False	True	0,750069
locf	locf	zscore	True	False	False	0,874485
locf	delete	minmax	True	True	True	0,876081
locf	delete	minmax	True	False	True	0,74993
locf	delete	minmax	True	False	False	0,871288

Tab. 8: Comparison of Preprocessing Options and their Impact on Accuracy.

MV	Outlier	Normalization	Limits	Smoothing	Filter	Accuracy
locf	delete	robust	True	True	True	0,878975
locf	delete	robust	True	False	True	0,752177
locf	delete	robust	True	False	False	0,871577
locf	delete	zscore	True	True	True	0,877106
locf	delete	zscore	True	False	True	0,751128
locf	delete	zscore	True	False	False	0,874404
delete	mean	minmax	True	True	True	0,869053
delete	mean	minmax	True	True	False	0,881291
delete	mean	minmax	True	False	True	0,74541
delete	mean	minmax	True	False	False	0,738213
delete	mean	minmax	False	True	True	0,869053
delete	mean	minmax	False	True	False	0,881291
delete	mean	minmax	False	False	True	0,74541
delete	mean	minmax	False	False	False	0,738213
delete	mean	robust	True	True	True	0,875088
delete	mean	robust	True	True	False	0,882806
delete	mean	robust	True	False	True	0,758888
delete	mean	robust	True	False	False	0,746336
delete	mean	robust	False	True	True	0,875088
delete	mean	robust	False	True	False	0,882806
delete	mean	robust	False	False	True	0,758888
delete	mean	robust	False	False	False	0,746336
delete	mean	zscore	True	True	True	0,87072
delete	mean	zscore	True	True	False	0,882768
delete	mean	zscore	True	False	True	0,74756
delete	mean	zscore	True	False	False	0,742932
delete	mean	zscore	False	True	True	0,87072
delete	mean	zscore	False	True	False	0,882768
delete	mean	zscore	False	False	True	0,74756
delete	mean	zscore	False	False	False	0,742932
delete	locf	minmax	True	True	True	0,873114
delete	locf	minmax	True	True	False	0,881077
delete	locf	minmax	True	False	True	0,760801
delete	locf	minmax	True	False	False	0,734923
delete	locf	minmax	False	True	True	0,873114
delete	locf	minmax	False	True	False	0,881077
delete	locf	minmax	False	False	True	0,760801
delete	locf	minmax	False	False	False	0,734923
delete	locf	robust	True	True	True	0,872961
delete	locf	robust	True	True	False	0,883158
delete	locf	robust	True	False	True	0,760495
delete	locf	robust	True	False	False	0,745365
delete	locf	robust	False	True	True	0,872961
delete	locf	robust	False	True	False	0,883158
delete	locf	robust	False	False	True	0,760495
delete	locf	robust	False	False	False	0,745365
delete	locf	zscore	True	True	True	0,869795
delete	locf	zscore	True	True	False	0,880848

Tab. 9: Comparison of Preprocessing Options and their Impact on Accuracy.

MV	Outlier	Normalization	Limits	Smoothing	Filter	Accuracy
mean	mean	robust	True	False	True	0,730538
mean	mean	robust	False	False	True	0,730538
mean	locf	robust	True	False	True	0,732228
delete	locf	zscore	True	False	True	0,753197
delete	locf	zscore	True	False	False	0,739459
delete	locf	zscore	False	True	True	0,869795
delete	locf	zscore	False	True	False	0,880848
delete	locf	zscore	False	False	True	0,753197
delete	locf	zscore	False	False	False	0,739459
delete	delete	minmax	True	True	True	0,868638
delete	delete	minmax	True	True	False	0,882152
delete	delete	minmax	True	False	True	0,751529
delete	delete	minmax	False	True	True	0,868638
delete	delete	minmax	False	True	False	0,882152
delete	delete	minmax	False	False	True	0,751529
delete	delete	robust	True	True	True	0,872606
delete	delete	robust	True	True	False	0,881483
delete	delete	robust	True	False	True	0,7592
delete	delete	robust	True	False	False	0,743647
delete	delete	robust	False	True	True	0,872606
delete	delete	robust	False	True	False	0,881483
delete	delete	robust	False	False	True	0,7592
delete	delete	robust	False	False	False	0,743647
delete	delete	zscore	True	True	True	0,869112
delete	delete	zscore	True	True	False	0,881374
delete	delete	zscore	True	False	True	0,751225
delete	delete	zscore	True	False	False	0,736855
delete	delete	zscore	False	True	True	0,869112
delete	delete	zscore	False	True	False	0,881374
delete	delete	zscore	False	False	True	0,751225
delete	delete	zscore	False	False	False	0,736855
mean	locf	robust	False	False	True	0,732228
mean	locf	zscore	True	False	True	0,725284
mean	locf	zscore	True	False	False	0,720189
mean	locf	zscore	False	False	True	0,725284
mean	locf	zscore	False	False	False	0,720189
delete	delete	minmax	True	False	False	0,732716
delete	delete	minmax	False	False	False	0,732716
mean	delete	minmax	True	True	False	0,91553
mean	delete	minmax	False	True	False	0,91553
locf	mean	minmax	True	True	False	0,919329
locf	mean	robust	True	True	False	0,919588
locf	mean	zscore	True	True	False	0,919532
locf	locf	minmax	True	True	False	0,919775

Tab. 10: Comparison of Preprocessing Options and their Impact on Accuracy.

C Appendix III: Missing Values and Outlier

Feature	MV (absolute)	MV (%)	Outlier (absolute)	Outlier (%)
ET_GazeLeftx	66959	7.92	5471	0.65
ET_GazeLefty	66959	7.92	9053	1.07
ET_GazeRightx	66959	7.92	5038	0.60
ET_GazeRighty	66959	7.92	8630	1.02
ET_CameraLeftX	66959	7.92	5896	0.70
ET_CameraLeftY	66959	7.92	5726	0.68
ET_CameraRightX	66959	7.92	4288	0.51
ET_CameraRightY	66959	7.92	4283	0.51
ET_ValidityLeft	66959	7.92	6276	0.74
ET_ValidityRight	66959	7.92	4283	0.51
Gaze X	372645	44.10	1402	0.17
Gaze Y	410975	49.16	6330	0.76
ET_PupilLeft	66472	8.15	6271	0.77
ET_PupilRight	66472	8.15	5854	0.72
ET_TimeSignal	66472	8.15	0	0
ET_DistanceLeft	66472	8.15	4760	0.58
ET_DistanceRight	66472	8.15	4923	0.60
Interpolated Gaze X	289954	35.22	1357	0.16
Interpolated Gaze Y	335282	40.73	6899	0.84
Interpolated Distance	276570	33.90	6088	0.75
Gaze Velocity	345022	42.28	10168	1.25
Gaze Acceleration	348406	42.70	9240	1.13
Fixation X	576159	70.61	655	0.08
Fixation Y	576159	70.61	3878	0.48
Fixation Start	576159	70.61	12	0.00
Fixation End	576159	70.61	12	0.00
Fixation Duration	576159	70.61	4750	0.58
Fixation Dispersion	587107	70.54	3677	0.44
Saccade Start	628512	77.03	64	0.01
Saccade End	628512	77.03	64	0.01
Saccade Duration	628512	77.03	3460	0.42
Saccade Amplitude	635406	77.87	4365	0.53
Saccade Peak Velocity	628512	77.03	3790	0.46
Saccade Peak Acceleration	630519	77.27	3805	0.47
Saccade Peak Deceleration	647087	79.30	3240	0.40
Saccade Direction	551053	80.76	0	0

Tab. 11: Overview of missing values and outlier per feature

SQLinked - A Hybrid Approach for Local and Database-Remote Program Execution

Florian Heinz [1] and Johannes Schildgen [1]

Abstract: When working with today's relational databases, there is usually a clear boundary between the database server and the application, that interfaces with the database system using the query language SQL. The concept of stored procedures allows to move complex parts of the business logic into the database server for various reasons, as, for instance, to reduce the latency of ELT processes that involve several database queries building on each other like distributing records into tables according to their attribute values. Creating and maintaining such stored procedures can be a challenging task, however. The idea pursued in this paper is to create a programming language, as well as a compilation and execution environment that allows the user to mark parts of the application code for being automatically compiled to and later be executed as a stored procedure in the database instead of the execution environment of the actual application. This blurs the border between database and application and provides a natural and maintenance-friendly way for offloading latency sensitive parts of the code to the database system.

Keywords: SQL databases, database programming, stored procedures, programming languages

1 Introduction

SQL is a standardized and powerful database query language allowing for complex queries, that shift parts the business logic from the actual application program to the database server as, for instance, data aggregation, window functions, complex mathematical transformations and more. But also more heterogeneous actions, possibly composed from several different types of database interactions like queries, updates and inserts, can be performed directly in the database by using *stored procedures*. This can be beneficial among other things for those reasons:

Reduced latency: Multiple commands that depend on each other are executed directly in the database avoiding network client induced latency
Low network traffic: Large amounts of data can be processed in the database system itself without transferring it over the network

One application for this is to improve an ELT process that, for instance, distributes data based on specific attribute values. In [CMS16] the benefits of reduced latency by avoiding unnecessary network round-trip-times are discussed and [Or24] provides a description how stored procedures in PL/SQL can be used to optimize database performance. Developing such stored procedures and their interplay with applications require significant effort in design, a break in program flow, logic and language, and is generally hard to maintain and keep in sync. This work proposes an approach for a programming language, compiler and virtual machine, that automatically converts specific parts of the program into a stored

1 OTH Regensburg, Databases, Galgenbergstr. 32, 93053 Regensburg, Germany,
florian.heinz@oth-regensburg.de, https://orcid.org/0000-0002-5380-5692;
johannes.schildgen@oth-regensburg.de, https://orcid.org/0000-0002-2450-0152

doi:10.18420/BTW2025-128

procedure, creates or updates it in the actual database system and automatically calls it when needed. This considerably eases the use of stored database procedures for efficiency and latency purposes. In this early version, the programmer can mark blocks manually that should be executed on the database system; however, the long-term vision is to have parts of the code automatically identified that benefit from being executed directly in the database system.

The remainder of the paper is structured as follows: First, an overview on related work for this subject is given. After that, a detailed example of an application that might benefit from this concept is presented. The next section shows, how the proof-of-concept implementation is designed. Then, an evaluation shows the performance benefits of this approach. Finally, an outlook on future work concludes the paper.

2 Related Work

A notable work in this realm is *WeBridge*. WeBridge [Hu24] specifically targets REST web applications. The database queries inside a program are transformed into a stored procedure, the runtime environment executes it and intercepts the traditional database driver calls accordingly. The road taken in our work is a more general approach. Instead of transferring only very specific statements of a stateless REST API to the database server, SQLinked translates whole and arbitrary parts of the program code to stored procedures that are completely executed by the database system and also complex control constructs like loops are supported. Other work on this topic includes Aggify [GPR20], which tries to replace a loop over a database cursor with an equivalent SQL query, that calls an aggregate function specifically crafted for the task at hand. In [GS08], multiple invocations of queries or stored procedures are optimized by automatically rewriting programs by replacing these calls with rewritten queries that are semantically equal. DBridge [Em17] tries to find relational operations in imperative code and rewrites it to perform these tasks directly in the database system using SQL queries.

3 Methodology

For demonstration purposes a procedure is used that involves several different database interactions that build on each other. First, the credentials are provided by the user. Then, the user record is fetched from the database. In case the user is unknown or the password is wrong, an error message is logged into a log table. If the login is successful, the time of last login in the user record is updated and an audit message of the successful login is written into the log table. In a naive approach, up to three database interactions are performed sequentially in the actual application: `SELECT`, `UPDATE` and `INSERT`. This incurs up to thrice the latency between the database server and the client. An assumed round trip time of 50ms would sum up to 150ms. Example code that implements an application this way is shown in Listing 1. The programming language *SQLinked* used in this example is a very simple scripting language that uses a static type system with type inference. Due to its simplicity it

```
1  dbconnect("user=sqlinked password=s3cr3t host=mdc1");
2  print("Username: "); login = getstring();
3  print("Password: "); pass  = getstring();
4  uid = 0;
5  ${  // Start of authentication code block
6  users = dbquery("SELECT * FROM accounts WHERE login = $1", login);
7  u = dbnext(users);
8  if (u) {
9      if (u["pass"] == pass) {
10             println("Login successful!");
11             dbquery("UPDATE accounts SET lastlogin=now() WHERE uid = $1", u["uid"]);
12             dbquery("INSERT INTO logs (action, comment) VALUES($1, $2)", "login", u["name"] + " logged in!");
13             uid = u["uid"];
14     } else {
15             println("Login failed!");
16             dbquery("INSERT INTO logs (action, comment) VALUES($1, $2)", "passwordfailed", u["name"] + " entered a wrong password!");
17             exit();
18     };
19 } else {
20     println("User does not exist!");
21     dbquery("INSERT INTO logs (action, comment) VALUES($1, $2)", "loginfailed", login + " does not exist!");
22     exit();
23 };
24 }; // End of authentication code block
25 print("Continuing with uid "); println(uid);
```

List. 1: Example of an application authenticating a user

is used in the proof-of-concept implementation, but in general, all languages with certain characteristics (e.g. a static type system) are suitable.

An alternative is to develop a stored procedure that performs the whole task with a single database command. The database management system *PostgreSQL* supports such procedures using the language *PL/pgSQL* [Gr24], which is largely inspired by *PL/SQL* [Or24] implemented in the Oracle DBMS. A function `authenticate(login text, password text)` could be implemented as a stored procedure that retrieves the account, logs if the user does not exist or the password is wrong and otherwise performs a successful login by logging it and updating the `lastlogin` field in the user record. This stored procedure can then be called from an application that is provided with some credential from user input like a console application, a web request or similar. Then, after successful login, the program flow continues with the determined userid.

The previously shown methods are either accompanied with higher latency or with manual effort for the stored procedure, which has to be created separately from the rest of the application, an interface has to be designed and changes in the program code may have to be evaluated if they also incur a change in the stored procedure on the database system. The idea to improve the situation is to simply mark parts of the code and the compiler will not create locally executed program parts for it, but compile it to a stored procedure and create stub code that calls this procedure with the current program context and reintegrates the results into the current program execution flow. In the example scenario above, lines 5 to 24 in Listing 1 is surrounded by a special block `${ ... }` called *dblock*, that triggers the described conversion into a stored procedure.

4 Programming Language and Compilation Process

In general, all programming languages fulfilling specific properties are suitable for the techniques described in this paper:

Paradigm: If the language follows the imperative programming paradigm it is more straightforward to convert it into PL/pgSQL [Gr24], but in general, all programming paradigmata are possible.

Type system: The language should be statically typed, because the data types of all involved variables have to be determined at compile time. This is necessary because PL/pgSQL [Gr24] is also a statically typed language and the type of all variables and formal function parameters have to be specified at compile time. The data types should ideally be widely compatible with the data types available in PL/pgSQL to avoid high conversion effort.

The proof-of-concept programming language *SQLinked* is part of the implementation to have full control over syntax and runtime environment. *SQLinked* supports the common control flow statements and data types and has a simple type inference system. *SQLinked* is normally compiled to bytecode that is executed in a virtual machine called *vm3*. For the support of stored procedures generation, an additional syntactic element, the *dblock*, was added. In Listing 1 the parts of the code inside the dblock surrounded by `${` and `}` are not compiled to bytecode, but to a PL/pgSQL stored procedure and stub code is generated that calls this stored procedure with the correct actual parameter values. On the other hand, the statements outside this block are translated regularly to vm3 bytecode for local execution.

```
CREATE OR REPLACE FUNCTION dblock_progbtw_global_1001 (INOUT login text, INOUT pass text, INOUT uid int) LANGUAGE plpgsql AS $$
DECLARE
 users refcursor;
 u record := row(NULL);
BEGIN
 OPEN users FOR EXECUTE 'SELECT * FROM accounts WHERE login = $1' USING login;
 FETCH users INTO u;
 IF u IS NOT NULL THEN
  IF u.pass = pass THEN
   RAISE NOTICE 'println/%', 'Login successful!';
   EXECUTE 'UPDATE accounts SET lastlogin=now() WHERE uid = $1' USING u.uid
   EXECUTE 'INSERT INTO logs (action, comment) VALUES($1, $2)' USING 'login', u.name || ' logged in!';
   uid = u.uid;
  ELSE
   RAISE NOTICE 'println/%', 'Login failed!';
   EXECUTE 'INSERT INTO logs (action, comment) VALUES($1, $2)' USING 'passwordfailed', u.name || ' entered a wrong password!';
   RAISE NOTICE 'exit/';
   RETURN;
  END IF;
 ELSE
  RAISE NOTICE 'println/%', 'User does not exist!';
  EXECUTE 'INSERT INTO logs (action, comment) VALUES($1, $2)' USING 'loginfailed', login || ' does not exist!';
  RAISE NOTICE 'exit/';
  RETURN;
 END IF;
END;
$$
```

List. 2: Stored procedure automatically created from designated code block

When the compilation process reaches the *dblock* the following happens:

External Variables: All used variables inside the block are checked if they are already declared outside the block, in which case these external variables are recorded and their type is determined. These variables become *INOUT* parameters to the stored procedure to pass the relevant context to the function. If the variables are modified during the actual procedure call, their updated value is passed back to the caller and the values in the calling virtual machine are modified accordingly. In Listing 1, the variables `login`, `pass` and `uid` are such external variables. Variables that are declared outside the *dblock* but not used inside are not considered.

Internal Variables: All other variables inside the *dblock* are *internal variables* and corresponding declarations have to be generated in the *DECLARE* block of the PL/pgSQL procedure. In the example Listing 1, the usage of variables `users` and `u` lead to `DECLARE users refcursor; u record := row(NULL);` in the generated stored procedure.

Operations and control statements: All operations and control statements inside the *dblock* part are compiled to PL/pgSQL sourcecode. In this step, the types of the involved variables and expressions has to be known, too. For example, the term `login + "␣does␣not␣exist!"` in the SQLinked source will be translated to `login || "␣does␣not␣exist!"` in PL/pgSQL to perform a string concatenation in the stored procedure, because `login` and the constant string literal are both of type string. For two integer variables, the + operator would have been used.

Database Queries: In *SQLinked*, database queries are sent using the function `dbquery(statement, param1, param2, ...)`. In normal code blocks, these function calls initiate a regular database query through the previously opened database connection. However, inside a *dblock* code block, no function call is performed but the database queries are tightly integrated into the stored procedure PL/pgSQL code. In Listing 2 line 6 the stored procedure code for the original *SQLinked* program in Listing 1 line 6 is shown. As the query result is assigned to the local variable *users*, a cursor is introduced and associated with this result. In contrast, Listing 2 line 11 does not have a result, hence *EXECUTE* is used.

Function calls: Other function calls inside the PL/pgSQL block are only supported in very specific cases. For one, the result of a function call is unavailable in the stored procedure and for another, the function call may not affect the global state of the program execution. In the proof-of-concept implementation only two function calls are supported: `println()` for emitting messages and `exit()` to halt program execution. The information about the function call and the actual parameters is transported with the *notify* mechanism through which the database client is informed about notices. In Listing 2 line 10 the message `println/Login successful!` is transported to the client, which performs the corresponding action by printing this message to the user. Line 17 on the other hand informs the execution environment to halt the program after the call to the stored procedure returns.

5 Evaluation

For evaluation purposes, two examples were chosen that are tested for correctness and performance. The setup consists of the application client host, that contains the actual runtime environment, as well as a database server running PostgreSQL 14.13 with a RTT of 50 ms between the hosts. Application 1 is the previous example in Listing 1 and Application 2 is a program that performs a transformation of an ELT process by iterating over the result set of a database query and performing action based on the values of each row, see Listing 3.

The evaluation results can be found in Table 1. Application 1 benefits latency-wise from the dblock execution mode because only a single query is issued to the database to call the previously generated stored procedure instead of three queries. The data transfer is similar to normal execution mode because the println-outputs are transferred to the client via the

```
dbconnect("user=sqlinked password=s3cr3t host=mdc1");
w = 0; a = 0; p = 0;

${
n = 0;
inc = dbquery("SELECT * FROM income"); // 10, 100, 500 or 1000 results
while (u = dbnext(inc)) {
    n = u["income"];
    if (n < 20000) {
        dbquery("INSERT INTO poor    (id, fname, lname, income) VALUES($1, $2, $3, $4)", u["id"], u["fname"], u["lname"], u["income"]); p++;
    } else { if (80000 < n) {
        dbquery("INSERT INTO wealthy (id, fname, lname, income) VALUES($1, $2, $3, $4)", u["id"], u["fname"], u["lname"], u["income"]); w++;
    } else {
        dbquery("INSERT INTO average (id, fname, lname, income) VALUES($1, $2, $3, $4)", u["id"], u["fname"], u["lname"], u["income"]); a++;
    }; };
};

};
print("Wealthy:"); println(w); print("Average:"); println(a); print("Poor:"); println(p);
```

List. 3: Example of an application looping over a result set

NOTICE mechanism. Latency and data transfer volume of Application 2 in *normal* mode largely depends on the number of rows in the *income* table and grows linearly with each loop iteration in normal execution mode. When activating the dblock execution mode, the latency is almost the same for 10 and 1000 iterations because no network communication latency is inferred. Also, the data transfer volume remains constant because the rows are not transmitted to the client.

	normal		dblock	
Application	timing	data transfer	timing	data transfer
1	165 ms	6632 bytes	56 ms	6543 bytes
2 (10 iterations)	962 ms	9991 bytes	54 ms	5650 bytes
2 (100 iterations)	5796 ms	49631 bytes	56 ms	5653 bytes
2 (500 iterations)	27283 ms	226783 bytes	62 ms	5654 bytes
2 (1000 iterations)	54177 ms	447831 bytes	65 ms	5656 bytes

Tab. 1: Evaluation results for SQLinked

6 Conclusion

In this paper we present an approach on how to make use of the benefits of stored procedures without having to care about details like the interface, switching the programming language and updating these stored procedures in the database when the program logic changes. The application can benefit from this with better maintainability, lower latency and lower network traffic. In future work, a bytecode interpreter could be installed on server side as an additional stored procedure language that can directly read and execute bytecode compiled from a programming language like *SQLinked* or similar. This eliminates the need for error-prone source-to-source compilation making the system more robust. Furthermore, static source-code analysis could automatically try to detect code blocks that particularly benefit from server side execution and perform this without explicit declaration. The code and example programs and data can be downloaded freely at `https://github.com/fwheinz/sqlinked`. The README file contains detailed instructions to reproduce the results presented in this work.

Bibliography

[CMS16] Cheung, Alvin; Madden, Samuel; Solar-Lezama, Armando: Sloth: Being Lazy Is a Virtue (When Issuing Database Queries). ACM Trans. Database Syst., 41(2):8:1–8:42, 2016.

[Em17] Emani, K. Venkatesh; Deshpande, Tejas; Ramachandra, Karthik; Sudarshan, S.: DBridge: Translating Imperative Code to SQL. In (Salihoglu, Semih; Zhou, Wenchao; Chirkova, Rada; Yang, Jun; Suciu, Dan, eds): Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017. ACM, pp. 1663–1666, 2017.

[GPR20] Gupta, Surabhi; Purandare, Sanket; Ramachandra, Karthik: Aggify: Lifting the Curse of Cursor Loops using Custom Aggregates. In (Maier, David; Pottinger, Rachel; Doan, AnHai; Tan, Wang-Chiew; Alawini, Abdussalam; Ngo, Hung Q., eds): Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020. ACM, pp. 559–573, 2020.

[Gr24] Group, The PostgreSQL Global Development: PL/pgSQL — SQL Procedural Language. 2024.

[GS08] Guravannavar, Ravindra; Sudarshan, S.: Rewriting procedures for batched bindings. Proc. VLDB Endow., 1(1):1107–1123, 2008.

[Hu24] Hu, Gansen; Wang, Zhaoguo; Tang, Chuzhe; Shen, Jiahuan; Dong, Zhiyuan; Yao, Sheng; Chen, Haibo: WeBridge: Synthesizing Stored Procedures for Large-Scale Real-World Web Applications. Proc. ACM Manag. Data, 2(1):64:1–64:29, 2024.

[Or24] Oracle: Database PL/SQL Language Reference. 2024.

Higher-Order SQL Lambda Functions

Maximilian E. Schüle[1]

Abstract: Model databases track the accuracy of models on pre-trained weights. The models are stored as executable code and extracted on deployment. Instead of extracting runnable code and data out of a database system, we propose higher-order SQL lambda functions for in-database execution. SQL lambda expressions have been introduced to let the user customise otherwise hard-coded data mining operators such as the distance function for k-means clustering. However, database systems parse lambda expressions during the semantic analysis, which does not allow for functions as arguments.

This paper proposes higher-order lambda functions that support the execution of functions from a table as input. Higher-order lambda functions expressing machine learning models allow data scientists to monitor the qualities over time and thus eliminate the need for any extraction step. This paper presents the conception of higher-order lambda functions and their embedding into relational algebra using a derived map operator. We further present the current prototype implementation on top of relational database systems and present preliminary results for data mining within SQL.

References

[SH24] Schüle, M. E.; Hornung, J.: Higher-Order SQL Lambda Functions. In: 40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024. IEEE, pp. 5622–5628, 2024, URL: https://doi.org/10.1109/ICDE60146.2024.00450.

[1] Universität Bamberg, An der Weberei 5, 96047 Bamberg, m.schuele@tum.de

doi:10.18420/BTW2025-129

Towards Complex Table Question Answering Over Tabular Data Lakes

Daniela Risis [1], Jan-Micha Bodensohn [1], Matthias Urban [1], and Carsten Binnig [1]

Abstract: Natural Language Interfaces for Databases (NLIDBs) are an interesting alternative to SQL since they empower non-experts to query data. However, they require this data to first be integrated into a database schema, causing high upfront data engineering and integration overheads. As such, Open Table Question Answering (OTQA) is promising since it allows directly querying tables in data lakes without first incorporating them into a relational schema. Many recent OTQA approaches combine Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs), where relevant tables are first retrieved from a data lake and then used as input to an LLM to answer the user query. In this paper, we take the first systematic step for investigating how LLMs paired with table retrievers can answer queries over private tabular data lakes. As a main finding, we see that even when tuning several parameters of this approach, current LLMs still fail to answer queries that focus on the simple extraction of individual cell values, let alone aggregate queries. Thus, they are far from the rich querying capabilities that NLIDB approaches offer today. To solve this, we point towards promising future work enabling complex question answering over tabular data lakes.

Keywords: Table Question Answering, Table Retrieval, Data Lakes, Large Language Models

1 Introduction

Natural Language Interfaces for Databases (NLIDBs) are an interesting alternative to SQL since they empower non-experts to query data [ART95]. However, an important prerequisite is that the data must first be stored in a relational database before it can be queried. This data preparation step often entails high manual overheads as the data must be explored, appropriate schemas designed, and the tables loaded into the database. Integrating all data into a relational schema is often not feasible for large table repositories like data lakes.

Open Table Question Answering (OTQA), on the other hand, answers natural language queries based on information stored in loose tables from large corpora such as data lakes [He21]. Therefore, it does not require the upfront effort of integrating all tables into a relational schema. Nevertheless, traditional OTQA approaches fall short of answering the complex SQL queries that NLIDBs can handle. For example, OTQA originally focused on extracting single-cell answers from individual tables and has only recently been extended to consider information from multiple tables [CZR24; Pa23].

[1] DFKI & Technical University of Darmstadt,
daniela.risis@stud.tu-darmstadt.de, https://orcid.org/0009-0004-8220-3051;
jan-micha.bodensohn@cs.tu-darmstadt.de, https://orcid.org/0000-0003-4884-0300;
matthias.urban@cs.tu-darmstadt.de, https://orcid.org/0000-0002-7418-6181;
carsten.binnig@cs.tu-darmstadt.de, https://orcid.org/0000-0002-2744-7836

doi:10.18420/BTW2025-130

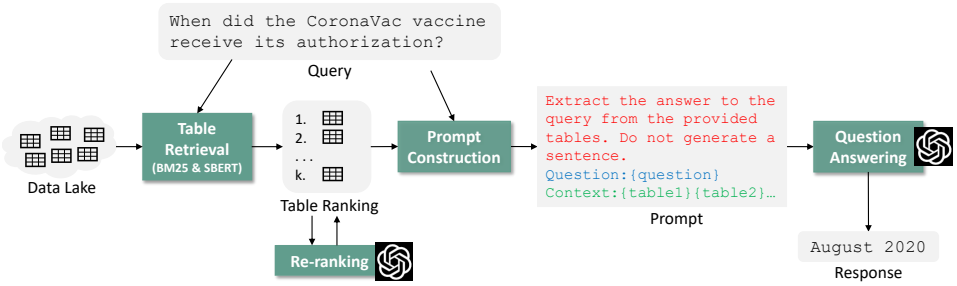


Fig. 1: Overview of our approach. Given a query, we first retrieve relevant tables from the data lake and optionally re-rank the top k tables using our novel re-ranking method. Afterward, we prompt the LLM to answer the query based on the retrieved tables.

Large Language Models (LLMs) have shown great promise for various table understanding tasks [DW24]. A common approach for OTQA combines an LLM with a table retriever in a Retrieval-Augmented Generation (RAG) setup, where the retrieved tables are used as context to ground the LLM-generated answer [He21]. What existing OTQA approaches have in common is that they are usually evaluated on queries about open-world knowledge from sources like Wikipedia [He21; Kw23; ZXS17]. Since these facts, styles of queries, and tables may have been part of the LLMs' training data, it remains unclear how LLMs perform on queries about the closed-domain knowledge typically stored in private data lakes.

Contributions. We take the first steps towards bridging the gap between NLIDBs and OTQA by examining how recent LLMs paired with table retrievers can answer questions over private tabular data lakes. Our contributions are: (1) We compare sparse and dense table retrieval methods, propose a novel LLM-based re-ranking strategy, and investigate various context representations and prompting strategies for the answer generation step. (2) Since existing OTQA datasets focus primarily on public world knowledge, we apply a question decontextualization approach similar to [Kw23] on the closed-domain AIT-QA dataset [Ka22], making it suitable for OTQA. (3) We extensively evaluate our approach on this adapted version of the AIT-QA dataset as well as the popular NQ-Tables dataset [He21] and compare it to closed-book and open-book baselines.

2 Methodology

Before we show the results of our extensive evaluation, we first describe the study setup. Following existing work on OTQA, we combine an LLM with a table retriever in a retriever-reader framework [He21]. As shown in Fig. 1, we first retrieve tables that may be relevant to the given query, which we then re-order using our LLM-based re-ranking strategy. Afterward, we construct a prompt based on the query and retrieved tables and use the LLM to generate the answer. In the following, we describe our approach in more detail.

2.1 Table Retrieval

Given a particular query, the table retrieval step ranks all tables in the data lake according to their relevance to the query. Based on previous research [Ma21; Wa22], we consider two retrieval methods: a sparse approach using BM25 [RZ09] and a dense approach based on SBERT [RG19]. Furthermore, we introduce a novel LLM-based re-ranking method to enhance the ordering of the top-ranked tables.

BM25. As a sparse retrieval method, we choose the bag-of-words-based BM25 ranking function [RZ09]. To adapt it to tables, we first linearize all tables as either JSON or CSV strings, also including metadata such as table names.

SBERT with cosine similarity. As a dense retrieval method, we use SBERT [RG19] to compute embeddings for the query and tables. To apply SBERT to tables, we again linearize the tables as either JSON or CSV strings, including the table metadata. Next, we rank the tables using a cosine similarity-based vector search.

LLM-based re-ranking. To further improve the order of the top-ranked tables, we introduce an LLM-based re-ranking approach similar to the method of [SSM24] for text data. The core idea is to represent each table as a set of LLM-generated queries that can be answered based on this table. Thus, we first instruct the LLM to generate multiple queries from a given table. Next, we use SBERT to compute embeddings for the generated queries and calculate their cosine similarities to the original query. We consider the maximum achieved cosine similarity score as the relevance of that table. Finally, we re-order the top k tables based on these new relevance scores.[2]

2.2 Table Question Answering

After retrieving relevant tables, we construct a prompt based on the query and retrieved tables and use the LLM to generate the answer. The prompt starts with a general instruction explaining the task and output format, followed by the query and relevant tables.

To construct the tabular context, we linearize the top n tables as either JSON or CSV strings including the table names.[2] Since LLMs are highly sensitive to the structure of the input prompt [Aj23], we compare three prompt templates that vary the position of the query, stating the query once in front of all tables (1), in front of each table (2), and once after all tables (3).

A second problem is caused by large tables that exceed the limited number of input tokens LLMs can process. To avoid dropping entire rows, we instead apply a token limit to all individual table cells, allowing us to mitigate cells containing long-form text. In case the context still exceeds the token limit, we cut it off after reaching the maximum input size.

[2] We choose $k = 10$ and $n = 3$ for our experiments.

3 Evaluation of LLM-based OTQA

To understand the gap between OTQA and NLIDBs, we evaluate our approach on two datasets. We start with an end-to-end evaluation before focusing on particular aspects like the table retrieval and context representation.

3.1 Datasets

As mentioned, OTQA approaches are typically evaluated on questions about open-world knowledge from sources like Wikipedia. To focus on their ability to answer questions over closed-domain tables in private data lakes, we prepare the following datasets:

NQ-Tables. NQ-Tables is an open-domain OTQA dataset of Google search queries with answers in Wikipedia tables [He21]. Our preliminary experiments have shown that LLMs can answer many queries even in a closed-book setup without provided context, which is unsurprising, since LLMs are often trained on scraped web data [RRS20]. To mitigate this data contamination, we partition the dataset into *easy* and *hard* queries based on whether they can be answered without context and focus our evaluation on the *hard* queries only.

AIT-QA. AIT-QA is a closed-domain table question answering dataset derived from corporate data in the airline industry [Ka22]. Contrary to NQ-Tables, it includes tables with complex structures (e.g. hierarchical headers), domain-specific terminology, and predominantly numerical data, making it particularly suitable for our evaluation. However, AIT-QA contains only table-question pairs, not originally intended for retrieval. To prepare it for the retrieval scenario, we apply a decontextualization approach inspired by [Kw23] that manually adds additional context information (e.g. airline names) to the tables.

3.2 Exp. 1: Table Question Answering

We start with an end-to-end evaluation that compares three scenarios: In the *closed-book* scenario serving as a lower baseline, the LLM must answer the question without any context [RRS20]. In the *oracle* scenario serving as an upper baseline, the LLM receives the question along with the ground truth table in CSV format. Finally, the *open-book* scenario corresponds to our RAG-based approach, which first retrieves relevant tables and then provides them in CSV format to the LLM to generate the answer.

We use two recent LLMs from OpenAI (`GPT-3.5-turbo-1106` and `GPT-4-0125-preview`) and evaluate with Exact Match (EM) accuracy and the token-based F1 measure from [Ra16].

Results. Tab. 1 shows that the table question answering results in the open-book setting still leave considerable room for improvement on both datasets. The performance gap to the oracle scenario indicates that incorrect answers are often caused by suboptimal tabular

	AIT-QA				NQ-Tables			
	EM		F1		EM		F1	
	G3.5	G4	G3.5	G4	G3.5	G4	G3.5	G4
Closed-book	0.02	0.02	0.02	0.04	0.03	0.04	0.16	0.19
Oracle (required table only)	0.82	0.91	0.82	0.91	0.53	0.52	0.67	0.67
Open-book: BM25	0.61	0.68	0.61	0.68	0.27	0.28	0.39	0.37
Open-book: BM25 + re-ranking	0.69	0.77	0.69	0.77	0.30	0.28	0.41	0.40

Tab. 1: Table question answering results leave considerable room for improvement.

	AIT-QA				NQ-Tables			
	Acc@1	Acc@3	Acc@5	Acc@10	Acc@1	Acc@3	Acc@5	Acc@10
BM25: JSON	0.46	0.80	0.87	0.94	0.25	0.43	0.53	0.59
BM25: CSV	0.43	0.75	0.85	**0.95**	0.26	0.45	0.53	0.61
BM25: CSV + re-ranking	**0.49**	**0.81**	**0.91**	**0.95**	0.27	0.45	0.51	0.60
SBERT: CSV	0.31	0.54	0.61	0.70	**0.32**	**0.52**	**0.60**	**0.71**

Tab. 2: Table retrieval accuracies show that the two datasets benefit from different retrieval methods.

contexts supplied by the table retrieval step. Therefore, providing the right table to the LLM is crucial to generate the correct answer. If the correct table is not retrieved, it is retrieved but not at rank 1 and thus de-prioritized by the LLM [Li23], or the additional irrelevant tables in the context distract the LLM, the answer generation is likely to fail.

Nevertheless, even the oracle setting does not achieve perfect table question answering results. While the particularly low performance on the NQ-Tables dataset may be explained by ambiguous questions with multiple answers, we speculate that the remaining gap on the clear-cut questions in the AIT-QA dataset is likely caused by its more complex table structures. Finally, the low performance in the closed-book setting shows that we have successfully mitigated any data contamination that would allow the LLMs to answer the queries based on their parametric knowledge.

To conclude, we find that table question answering results are highly dependent on the provided tabular context and thus primarily driven by the table retrieval performance.

3.3 Exp. 2: Table Retrieval and Prompt Construction

Our second experiment provides ablations for the table retrieval and prompt construction.

Table retrieval. Tab. 2 shows the retrieval accuracies for various retrieval approaches. First, we observe consistently worse results on NQ-Tables compared to AIT-QA, which we attribute to the much larger size of the NQ-Tables corpus (170k tables vs. 113 tables). Furthermore, the two datasets generally benefit from different retrieval approaches. For example, while

BM25 outperforms SBERT on AIT-QA, SBERT wins on NQ-Tables. A potential explanation is that the word-based BM25 benefits from the limited domain vocabulary of the tables and questions in AIT-QA. On the other hand, the tables in NQ-Tables are larger and contain mostly textual content, thus displaying a higher variability of the vocabulary. Finally, we find that whereas CSV representations outperform JSON on NQ-Tables, JSON performs better than CSV on AIT-QA for Acc@1,3,5.

Re-ordering the top 10 tables retrieved by BM25 using our LLM-based re-ranking strategy yields consistent improvements on AIT-QA. However, the re-ranking only leads to small benefits on NQ-Tables. Our error analysis shows that there are several reasons for this, including multiple highly-similar tables leading to similar generated questions as well as long textual cell contents leading to questions that distract from the table's overall topic.

Prompt construction. We further compare the different prompt templates discussed in Sect. 2 that vary the positions of the query and tabular context. Our experiments show that while AIT-QA benefits slightly from template (2) placing the query in front of each table, the differences on NQ-Tables are negligible.

4 Outlook

We have presented a detailed analysis of various methods for RAG-based OTQA and proposed a novel LLM-based table re-ranking method. Despite tuning several parameters, current LLMs still fail to answer even simple queries over private data lakes, highlighting that there is still a substantial gap between NLIDBs and OTQA.

Since our experiments show that LLMs often fail to answer questions even if the required tabular context is provided, future work should focus on extending the query execution capabilities of LLMs. Potential solutions to this problem include fine-tuning LLMs to execute queries [Pa23], splitting larger tables into smaller segments [BB24; Le24], and using LLMs to derive query execution plans [Bi24; Pa24; UB23].

Our experiments also demonstrate that the question answering performance depends strongly on providing the right tables to the LLM. Therefore, future work should improve the effectiveness of the table retrieval step. This is especially important for questions that require aggregating information from multiple tables [CZR24], which is a common task that NLIDBs must handle. Recently, novel OTQA datasets have appeared that are designed for a multi-table retrieval setting [Qi24; Wu24; Wu25]. On the other hand, approaches emerge to better solve the retrieval problem [CZR24] or better align the question with the organization of the data collection [Ch25]. In addition, generative information retrieval proposed in the domain of text retrieval could be explored for table retrieval [Ze24].

Finally, we want to stress the importance of evaluating OTQA approaches on datasets resembling domain-specific private tabular data lakes to avoid data contamination and ensure that improvements actually translate into practice.

Acknowledgments

This work has been supported by the BMBF and the state of Hesse as part of the NHR Program and the HMWK cluster project 3AI. It was also partially funded by the LOEWE Spitzenprofessur of the state of Hesse. We also thank DFKI Darmstadt and hessian.AI for their support.

References

[Aj23] Ajith, A.; Pan, C.; Xia, M.; Deshpande, A.; Narasimhan, K.: Instructeval: Systematic evaluation of instruction selection methods. arXiv preprint arXiv:2307.00259, 2023.

[ART95] Androutsopoulos, I.; Ritchie, G.; Thanisch, P.: Natural language interfaces to databases – an introduction. Natural Language Engineering 1 (1), pp. 29–81, 1995, DOI: 10.1017/S135132490000005X.

[BB24] Bodensohn, J.-M.; Binnig, C.: Rethinking Table Retrieval from Data Lakes. In: Proceedings of the Seventh International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. aiDM '24, Association for Computing Machinery, New York, NY, USA, pp. 1–5, 2024, ISBN: 979-8-4007-0680-6, DOI: 10.1145/3663742.3663972, URL: https://dl.acm.org/doi/10.1145/3663742.3663972, visited on: 06/19/2024.

[Bi24] Biswal, A.; Patel, L.; Jha, S.; Kamsetty, A.; Liu, S.; Gonzalez, J. E.; Guestrin, C.; Zaharia, M.: Text2SQL is Not Enough: Unifying AI and Databases with TAG. CoRR abs/2408.14717, 2024, DOI: 10.48550/ARXIV.2408.14717, arXiv: 2408.14717, URL: https://doi.org/10.48550/arXiv.2408.14717.

[Ch25] Chen, P. B.; Zhang, Y.; Cafarella, M.; Roth, D.: Can we Retrieve Everything All at Once? ARM: An Alignment-Oriented LLM-based Retrieval Method. arXiv preprint arXiv:2501.18539, 2025.

[CZR24] Chen, P. B.; Zhang, Y.; Roth, D.: Is Table Retrieval a Solved Problem? Exploring Join-Aware Multi-Table Retrieval. In (Ku, L.-W.; Martins, A.; Srikumar, V., eds.): Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand, pp. 2687–2699, 2024, DOI: 10.18653/v1/2024.acl-long.148, URL: https://aclanthology.org/2024.acl-long.148.

[DW24] Dong, H.; Wang, Z.: Large language models for tabular data: Progresses and future directions. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pp. 2997–3000, 2024.

[He21] Herzig, J.; Müller, T.; Krichene, S.; Eisenschlos, J.: Open Domain Question Answering over Tables via Dense Retrieval. In (Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; Zhou, Y., eds.): Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Online, pp. 512–519, 2021, DOI: 10.18653/v1/2021.naacl-main.43, URL: https://aclanthology.org/2021.naacl-main.43.

[Ka22] Katsis, Y.; Chemmengath, S.; Kumar, V.; Bharadwaj, S.; Canim, M.; Glass, M.; Gliozzo, A.; Pan, F.; Sen, J.; Sankaranarayanan, K., et al.: AIT-QA: Question Answering Dataset over Complex Tables in the Airline Industry. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track. Pp. 305–314, 2022.

[Kw23] Kweon, S.; Kwon, Y.; Cho, S.; Jo, Y.; Choi, E.: Open-WikiTable : Dataset for Open Domain Question Answering with Complex Reasoning over Table. In (Rogers, A.; Boyd-Graber, J.; Okazaki, N., eds.): Findings of the Association for Computational Linguistics: ACL 2023. Association for Computational Linguistics, Toronto, Canada, pp. 8285–8297, 2023, DOI: 10.18653/v1/2023.findings-acl.526, URL: https://aclanthology.org/2023.findings-acl.526.

[Le24] Lee, W.; Kim, K.; Lee, S.; Lee, J.; KIm, K. I.: Piece of Table: A Divide-and-Conquer Approach for Selecting Sub-Tables in Table Question Answering. arXiv preprint arXiv:2412.07629, 2024.

[Li23] Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P.: Lost in the Middle: How Language Models Use Long Contexts. CoRR abs/2307.03172, 2023, DOI: 10.48550/ARXIV.2307.03172, arXiv: 2307.03172, URL: https://doi.org/10.48550/arXiv.2307.03172.

[Ma21] Ma, X.; Sun, K.; Pradeep, R.; Lin, J.: A Replication Study of Dense Passage Retriever. CoRR abs/2104.05740, 2021, arXiv: 2104.05740, URL: https://arxiv.org/abs/2104.05740.

[Pa23] Pal, V.; Yates, A.; Kanoulas, E.; de Rijke, M.: MultiTabQA: Generating Tabular Answers for Multi-Table Question Answering. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Toronto, Canada, pp. 6322–6334, 2023, DOI: 10.18653/v1/2023.acl-long.348, URL: https://aclanthology.org/2023.acl-long.348.

[Pa24] Patel, L.; Jha, S.; Guestrin, C.; Zaharia, M.: Lotus: Enabling semantic queries with llms over tables of unstructured and structured data. arXiv preprint arXiv:2407.11418, 2024.

[Qi24] Qiu, Z.; Peng, Y.; He, G.; Yuan, B.; Wang, C.: TQA-Bench: Evaluating LLMs for Multi-Table Question Answering with Scalable Context and Symbolic Extension. arXiv preprint arXiv:2411.19504, 2024.

[Ra16] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P.: SQuAD: 100,000+ Questions for Machine Comprehension of Text. In (Su, J.; Duh, K.; Carreras, X., eds.): Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, pp. 2383–2392, 2016, DOI: 10.18653/v1/D16-1264, URL: https://aclanthology.org/D16-1264, visited on: 12/14/2024.

[RG19] Reimers, N.; Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Pp. 3982–3992, 2019.

[RRS20] Roberts, A.; Raffel, C.; Shazeer, N.: How Much Knowledge Can You Pack Into the Parameters of a Language Model? In (Webber, B.; Cohn, T.; He, Y.; Liu, Y., eds.): Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, pp. 5418–5426, 2020, DOI: 10.18653/v1/2020.emnlp-main.437, URL: https://aclanthology.org/2020.emnlp-main.437.

[RZ09] Robertson, S.; Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Information Retrieval 3 (4), pp. 333–389, 2009.

[SSM24] Saha, B.; Saha, U.; Malik, M. Z.: Advancing Retrieval-Augmented Generation with Inverted Question Matching for Enhanced QA Performance. IEEE Access, 2024.

[UB23] Urban, M.; Binnig, C.: CAESURA: Language Models as Multi-Modal Query Planners. arXiv preprint arXiv:2308.03424, 2023.

[Wa22] Wang, Z.; Jiang, Z.; Nyberg, E.; Neubig, G.: Table Retrieval May Not Necessitate Table-specific Model Design. In: Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI). Pp. 36–46, 2022.

[Wu24] Wu, X.; Yang, J.; Chai, L.; Zhang, G.; Liu, J.; Du, X.; Liang, D.; Shu, D.; Cheng, X.; Sun, T., et al.: Tablebench: A comprehensive and complex benchmark for table question answering. arXiv preprint arXiv:2408.09174, 2024.

[Wu25] Wu, J.; Yang, L.; Li, D.; Ji, Y.; Okumura, M.; Zhang, Y.: MMQA: Evaluating LLMs with Multi-Table Multi-Hop Complex Questions. In: The Thirteenth International Conference on Learning Representations. 2025, URL: https://openreview.net/forum?id=GGlpykXDCa.

[Ze24] Zeng, H.; Luo, C.; Jin, B.; Sarwar, S. M.; Wei, T.; Zamani, H.: Scalable and effective generative information retrieval. In: Proceedings of the ACM on Web Conference 2024. Pp. 1441–1452, 2024.

[ZXS17] Zhong, V.; Xiong, C.; Socher, R.: Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning, arXiv:1709.00103 [cs], 2017, URL: http://arxiv.org/abs/1709.00103, visited on: 03/06/2024.

The Kosmosis Use Case of Crypto Rug Pull Prevention by an Incrementally Constructed Knowledge Graph

Philipp Stangl [1] and Christoph P. Neumann [2]

Abstract: Current methods to prevent crypto asset fraud are based on the analysis of transaction graphs within blockchain networks. While effective for identifying transaction patterns indicative of fraud, it does not capture the semantics of transactions and is constrained to blockchain data. Consequently, preventive methods based on transaction graphs are inherently limited. In response to these limitations, we propose the Kosmosis approach, which aims to incrementally construct a knowledge graph as new blockchain and social media data become available. During construction, it aims to extract the semantics of transactions and connect blockchain addresses to their real-world entities by fusing blockchain and social media data in a knowledge graph. This enables novel preventive methods against rug pulls as a form of crypto asset fraud. To demonstrate the effectiveness and practical applicability of the Kosmosis approach, we examine a series of real-world rug pulls. Through this case, we illustrate how Kosmosis can aid in identifying and preventing such fraudulent activities by leveraging the insights from the constructed knowledge graph.

Keywords: blockchain, cyber fraud, rug pull, knowledge graphs, pseudonyms, traceability.

1 Introduction

Crypto assets are digital assets that use distributed ledger technology, such as blockchain, to prove ownership and maintain a decentralized and public ledger of all transactions. There are distinct types of assets, each with unique characteristics and use cases. Cryptocurrencies, like Bitcoin [Na08], are the most well-known form. They function as digital currencies and are used for storing or transferring monetary value. Fungible Tokens (FTs), another type of crypto asset, are interchangeable units representing various utilities or assets within a blockchain ecosystem. These tokens often play a vital role in Decentralized Finance (DeFi) protocols and can represent anything from voting rights to a currency within a project ecosystem. Lastly, Non-Fungible Tokens (NFTs) are unique digital assets that prove ownership and authenticity of digital or real-world assets [Al23]. Unlike cryptocurrencies and FTs, each NFT has a distinct value and cannot be exchanged on a one-to-one basis with other tokens. In the rapidly evolving landscape of crypto assets, the incidence of illicit activities has surged. Chainalysis, a leading blockchain analytics firm, reported that illicit transaction volume rose for the second consecutive year, reaching an all-time high of $20.6 billion in illicit activity [Ch23a]. Since the rise of DeFi in 2020, followed by NFTs in 2021, rug pulls have become a major fraud scheme in terms of amount stolen and frequency

1 Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany,
philipp.stangl@fau.de, https://orcid.org/0009-0007-4179-2365
2 Ostbayerische Technische Hochschule Amberg-Weiden, Amberg, Germany,
c.neumann@oth-aw.de, https://orcid.org/0000-0002-5936-631X

doi:10.18420/BTW2025-131

[SAS23]. Thus, rug pulls pose a significant risk to investors and undermine the integrity of the crypto asset sector.

The predominant approach for identifying patterns indicative of fraudulent activity is the transaction graph analysis within blockchain networks [Bé21; Kh22], [Hu21, pp. 21–24]. However, this approach presents two key challenges. Firstly, the transacting parties are pseudonymous and only their blockchain addresses are publicly known. This means that, although the transactions of a specific address can be tracked, linking that address to a real-world entity can be challenging since this approach is limited to information or patterns observable in blockchain data. Secondly, this approach is only concerned with the following aspects of a transaction: 1) The transferred asset, 2) the quantity, and 3) the sender and receiver. However, the semantics of a transaction, such as what happened in a transaction that caused the assets to get transferred, is not covered. Thereby limiting the depth of analysis that can be conducted on crypto asset movements.

Knowledge Graphs (KGs) [Ho22] are increasingly recognized as a powerful means to integrate fragmented knowledge from heterogeneous data sources into a graph of data to facilitate semantic querying (e. g., [Ch23b; Ha22b]) and reasoning (e. g., [NFL10]). A KG provides a holistic view for identifying patterns and hidden connections indicative of fraudulent activities in a highly connected dataset [Zh21]. The KG consists of semantically described entities, each with a unique identifier, and relations among those entities using an ontological representation [FW16; Ho23]. Their open world assumption allows for the continual integration of new data. By leveraging these capabilities, KGs can enhance crypto asset fraud analysis and aid in predicting future fraudulent activities.

2 Kosmosis Objectives

This section outlines the objectives of Kosmosis beginning with the primary objective that investigates the potential of a KG in identifying and alerting users before they interact with projects linked to known scammers, addressing a critical need for security and trust in blockchain ecosystems. Following that, we explore the technical implications.

Objective 1: How can the KG identify and aid in alerting users before interacting with a rug pull project?

With the rise in illicit activities in the crypto asset market, especially rug pulls, there is a pressing need for effective means to detect and prevent fraudulent activities. Kosmosis aims to integrate fragmented knowledge from blockchains like Ethereum, social media like $\mathbb{X}$[3], into one unified KG, enabling semantic querying and reasoning over a graph of entities and the relationships among them. The KG could serve as a knowledge base for a real-time alerting system, warning users of potential risks associated with certain projects or individuals.

3 $\mathbb{X}$ is the platform formerly known as Twitter.

Objective 2: How to incrementally construct the KG from heterogeneous data sources?

It is imperative to establish a pipeline capable of integrating updates into the KG in both batch- and streaming-like manner. Thereby, maintaining high data freshness by ensuring that the KG consistently reflects the most up-to-date information from the blockchain and other sources. This approach should not entail a complete reconstruction of the KG, but rather concentrate on integrating new information, avoiding the reprocessing of data that is already incorporated.

Objective 3: How to extract the semantics of blockchain transactions?

Transaction graphs commonly only display transactions with asset transfers and provide answers to questions such as "what" assets were transferred and "where" were they transferred to. Understanding transactions semantically is vital in uncovering sophisticated fraudulent schemes that might otherwise go unnoticed. Kosmosis addresses this gap by extracting the semantics of transactions, providing answers to "why" and "how" assets were transferred in a transaction. This extraction of semantic information is primarily achieved through decoding the input data of a transaction using the Application Binary Interface (ABI) of smart contracts a transaction interacts with.

3 Background

Blockchain technology[4] is based on the principles of immutability, decentralization, transparency, and cryptographic security and has seen various applications in recent years. For instance, in the financial sector (e. g., [Na08; Wo24]), or supply chain management (e. g., using a single blockchain [Wa19], or using multiple, interoperable blockchains [SN23; St22]). Smart contract platforms represent a subset of blockchains that enable the development of decentralized applications through smart contracts. This section outlines the key concepts of Ethereum, as an example for smart contract platforms, that are essential for the following sections of this work, such as smart contracts, their execution environment, and account-based accounting. For the record-keeping of transactions, blockchains utilize an accounting model. Compared to other blockchains, such as the equally well-known Bitcoin [Na08] blockchain that uses the Unspent Transaction Output (UTXO) model, or its successor the extended UTXO [Ch20a] utilized by the Cardano [Ho17] blockchain, Ethereum [Wo24] employs the account-based accounting model. The account-based model can be best understood through the analogy of a bank account. This approach mirrors how a banking account operates, thus, it tracks the inflow and outflow of funds, thereby reflecting the current balance, the account-based model in Ethereum maintains a state that records the balance of Ether. This stateful nature of the model ensures that at any given moment, the system can accurately reflect the total amount of Ether held in each account, offering an up-to-date view of account balances within Ethereum.

4 As additional background, we did some previous work on correlating Reddit data with traditional stock market data in [Ba22] as well as analyzing Twitter data with SPARQL [Ha22a].

3.1 Blockchain Data Structure

A blockchain is a data structure whose elements called blocks are linked together to form a chain of blocks, as depicted in Figure 1 [Zh17]. Each block comprises a body containing a set of transactions and a header referencing the unique identifier (i. e., hash) of its immediate predecessor. Transactions typically involve the transfer of assets between addresses, which are unique alphanumeric strings specifying the sender and recipient. A smart contract platform like Ethereum extends this structure through smart contracts—executable source codes that enforce the terms and conditions of particular agreements, to facilitate the development of decentralized applications [Ma23]. Once deployed on the blockchain, the smart contract is assigned an address where the code resides and cannot be altered or tampered with. Developers can create and manage tokens conforming to standards like ERC-20 (FT) [VB15] or ERC-721 (NFT) [En18], and interact with the contract via an ABI, which defines the functions and data structures exposed by the contract and a standardized way to encode/decode data. Ethereum's native cryptocurrency is Ether (ETH), and it leverages the Ethereum Virtual Machine (EVM) to execute contract code. Smart contracts, primarily written in Solidity, are compiled into EVM bytecode. A contract ABI bridges high-level code and low-level bytecode, while an EVM disassembler can reverse the bytecode into a more readable format. Finally, Externally Owned Accounts (EOAs), controlled by private keys, can initiate transactions to transfer crypto assets or invoke smart contract functions.

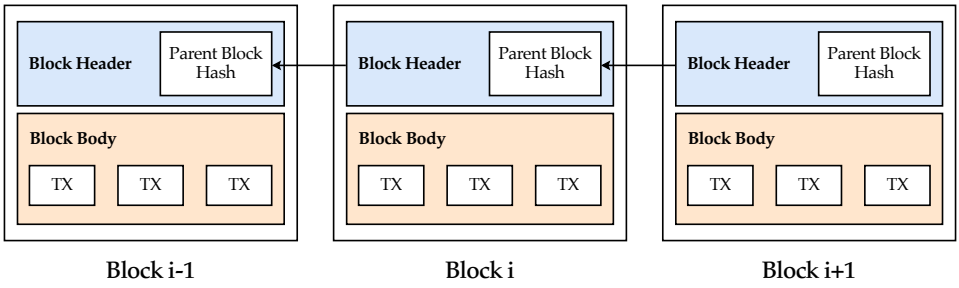


Fig. 1: Schematic representation of the blockchain data structure. Adapted from Zheng et al. [Zh17].

3.2 Token Minting

The process of creating new tokens is called token minting. Token mints are often governed by predefined rules or algorithms embedded within smart contracts. Fungible tokens are typically minted by the creator either at the inception of the project or progressively over time. In contrast, NFT minting involves other individuals besides the token creator, commonly termed as token minters. They engage by invoking a specific function within a smart contract, in the ERC-721 token standard, called mint. This action results in an increase in the supply of the NFTs and simultaneously assigns these minted tokens to the blockchain address of

the minter. The mechanism of minting NFTs often involves utilizing a dedicated minting website. Here, prospective minters or investors are required to invest a predetermined amount, as set by the creator, to initiate the minting process. This investment grants them the ability to mint one or multiple NFTs, depending on the terms set forth in the smart contract. This process not only facilitates the creation of new NFTs but also serves as a means of transferring ownership directly from the creator to the NFT minter.

3.3 Rug Pull Detection Methods

Two primary methods have been employed in the past to detect rug pulls: smart contract code analysis and graph-based methods. Smart contract code analysis involves a thorough examination of the contract's code to extract and analyze the semantic behavior of transactions. For instance, [Zh24] utilizes smart contract code analysis to reveal potential vulnerabilities and fraudulent patterns within the contracts. By dissecting the code, their proposed method, dubbed "Tokeer", can identify suspicious patterns and functions that might indicate a predisposition to rug pull scams. Graph theories and graph-based data mining methods are applicable for discovering information in blockchain network graphs, because blockchain transactions can be easily structured into graphs [Hu21]. Elmougy; Liu [EL23] identified three types of graphs, applicable to any blockchain network: *money flow transaction graphs* visualize the asset flow over time, *address-transaction graphs* showing flow of an asset across transactions and addresses, and *user entity graphs* that clusters the graph for potential linking of addresses controlled by the same user, to deanonymize their identity and purpose. To detect rug pulls, graph-based approaches use network embedding methods to automatically extract features from the blockchain network (e. g., [Ch20b]) using a *graph convolutional network*.

4 Rug Pulls and Serial Fraudsters

A rug pull can be categorized as a scam, i. e., the victim authorizes the transaction. This type of scam is typically carried out in five stages, according to [SAS23]: (1) Project creation with roadmap and total supply of tokens (optional), (2) pre-mint hype, (3) set token mint price, (4) token mint, accumulation of more capital and increase in popularity, and finally (5) the creators cash out, abandon the project, and leave the investors defrauded. To attract users and investments for rug pulls, Sharma et al. [SAS23] suggest the involvement of individuals or groups that possess substantial technical skills and knowledge of blockchain technology and demonstrate a proficiency in marketing techniques. This specific use case is particularly relevant given the findings in [SAS23] and [MAD22]: Mazorra et al., who analyzed ERC-20 tokens listed on decentralized exchanges in their 2022 study, labeled 97.7% out of 27,588 analyzed tokens as rug pulls [MAD22]. Likewise, Sharma et al. analyzed NFTs and identified a cluster of 168 NFTs associated with what they termed the "Rug-Pull Mafia," a group of creators responsible for orchestrating multiple and repeated rug pulls

[SAS23]. There is a growing trend in both the frequency and the financial impact of crypto rug pulls and scams [Mo23].

5 The Use Case of Rug Pull Prevention

To illustrate the vision of Kosmosis-enabled rug pull prevention methods, this section introduces a hypothetical user story centered around a character we name Bob, a crypto market participant. The Kosmosis user story is designed to provide a relatable perspective on how individuals like Bob are affected by such fraudulent activities. The story of Bob, while fictional, is grounded in a series of real-world rug pulls that took place in 2021. All rug pulls were carried out by the same fraudulent NFT creator and Twitter user known as Homer_eth. In Section 5.3, we outline how the series of rug pulls experienced by Bob might have unfolded differently had he been equipped with a Kosmosis-enabled fraud prevention mechanism at the time.

5.1 Past User Story

In the span of two months, from October to November 2021, a fraudulent NFT creator and 𝕏 user known as Homer_eth executed five different NFT project rug pulls within two months, accumulating over $2.8 million in profits. Table 1 provides an overview of Homer_eth's rug pull projects, each with launch date and the estimated profit. The basis of this user story is the transaction graph depicted in Fig. 2 that provides a simplified visualization of the transaction flow across multiple NFT projects linked to Homer_eth. It highlights key components, including EOA Nodes (Externally Owned Accounts), which represent the multiple wallet addresses of the rug puller, and Deployer Nodes (Smart Contract Creators), with the 0xc8a6 address being the deployer for multiple fraudulent contracts. The links between addresses are established through various transaction, such as mint transactions (e.g., mintReaper, mintBananaChips), which indicate purchases; fund transfers (e.g., Transfer 65.61 ETH to 0xc8a6), showing proceeds flowing to personal wallets or exchanges; and contract deployments (e.g., Deploy Ether Reapers). The transaction graph makes a critical indication of fraudulent intent visible. Instead of using a multisig treasury or project contract, funds were immediately funneled to a single address controlled by the rug puller.

Bob's story begins with a common enthusiasm for the burgeoning world of NFTs. His journey into the NFT market is marked by excitement and optimism, spurred by the success stories he sees online. Homer_eth, an NFT creator and 𝕏 user, has caught the attention of many like Bob by sharing his NFT projects on 𝕏. His first NFT collection was *Ether Bananas*, consisting of 750 NFTs, was launched on October 7, 2021. Only four days later, on October 11, Homer_eth continued with the release of *Ether Monkeys*, followed by the release of *Zombie Monkeys*. The buzz around Homer_eth's projects, especially *Ether*

Tab. 1: Rug Pull Projects by Homer_eth

Project Name	Launch Date	Estimated Profit
Ether Bananas	10/07/2021	$125k
Ether Monkeys	10/11/2021	$1.77m
Zombie Monkeys	10/15/2021	$413k
Ether Reapers	10/20/2021	$282k
ETH Banana Chips	11/23/2021	$208k

Monkeys, which promised additional utility through a casino to gamble and a decentralized autonomous organization to govern the NFTs, according to [Za22], draws Bob into the fray. Being relatively new to the NFT market, Bob views this as an opportunity not to be missed. Bob bought his first NFT from Homer_eth, an *Ether Reapers*, and with that purchase, he was no longer just a bystander; he was now an active participant in Homer_eth's growing community.

Bob's involvement in the community deepened over time. He engaged in discussions, shared his excitement with fellow members, and reveled in the rumors of more NFT launches in the future. His commitment paid off when he earned himself a whitelist spot that allows Bob to mint the upcoming NFT project *ETH Banana Chips* by Homer_eth. Convinced of its potential, Bob didn't hesitate to mint an *ETH Banana Chips* NFT when the opportunity arose. With a click to confirm the transaction in his browser wallet (e. g., MetaMask [Me23]), Bob became the proud owner of an *ETH Banana Chips* NFT, unaware of the underlying risks associated with his investment.

However, the reality of the situation was far from the optimistic scenario Bob had envisioned. Unknown to him, since Bob had a limited understanding of blockchain transactions, the proceeds from the *Ether Reapers* mint were not being locked in the smart contract for future development as promised. Instead, they were directly funneled into Homer_eth's deployer address. From there, Homer_eth will later transfer those mint proceeds either to his next

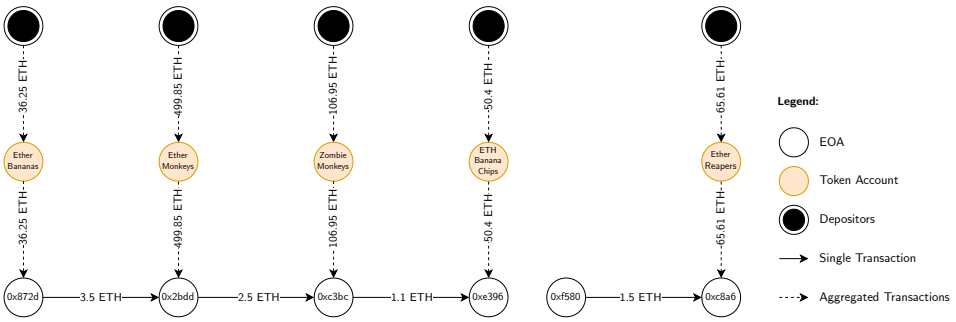


Fig. 2: Simplified transaction graph of Homer_eth's NFT rug pulls.

deployer address for a future rug pull or to an exchange in order to pay out his profits made from rug pulling the projects.

After the launch of *ETH Banana Chips*, a tense silence enveloped the community. For months, there was no news from Homer_eth, no updates on the project, leaving everyone to wonder about the future. It wasn't until March 2022, that Homer_eth broke the silence with the announcement of one last NFT project, dubbed *Froggy Frens*. However, due to backlash from the community, Homer_eth deleted his 𝕏 account and vanished [Za22].

5.2 Kosmosis Extension

The basis of the extended user story is the Kosmosis KG, depicted for this specific user story in Figure 3. This KG is a direct enhancement over the basic transaction graph from Figure 2 as it was discussed in the previous subsection. The enhancement comprises semantically annotated edges and the incorporation of data from the social media platform 𝕏.

To enhance the interpretability of raw blockchain transactions, we introduce semantic annotations in our knowledge graph. This process involves using the Application Binary Interface (ABI) of smart contract transactions. Transactions interacting with NFT minting contracts contain function calls embedded in input data. Using the ABI, we extract function names (e.g., mintMonkey, mintBananaChips) and parameters. This enables labeling edges

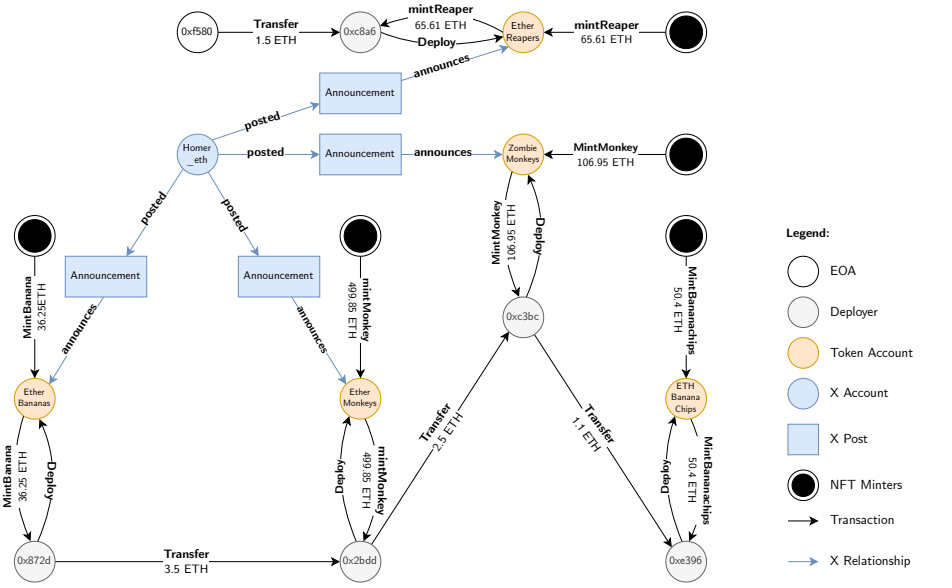


Fig. 3: Knowledge graph of Homer_eth's NFT rug pulls, constructed using Kosmosis.

as minting transactions rather than generic asset transfers. Transaction type classification is done by categorizing transfers into value transactions, such as *mintMonkey* and *Transfer*, and non-value transactions, like contract deployments denoted as *Deploy*. These semantics allow describing (i. e., tagging) sender and receiver addresses as NFT minter (previously depositor) and deployer (previously EOA) addresses.

The data integrated from platform 𝕏 enriches the KG with detailed information about user accounts, labeled as *X Account*, and specific announcements or posts, referred to as *X Post*. This integration facilitates a deeper understanding of the context and relationships surrounding these rug pulls. For instance, the KG can establish a connection between previously unrelated entities, such as the deployer address *0xc8a6* and the user Homer_eth. This connection is made through a social media announcement in which Homer_eth claims to have created the Ether Bananas project, as well as through semantic annotation, which identifies *0xc8a6* as the deployer address of the Ether Bananas smart contract.

5.3 Future User Story

In an alternative scenario where Bob would have had access to Kosmosis-enabled rug pull prevention, his journey in the NFT market would have been safer, beginning with his initial transaction to purchase an *Ether Reapers* NFT.

As soon as Bob initiated his transaction, the rug pull prevention mechanism would have accessed the KG, to analyze the rug pull risk of the contract. Based on the integrated knowledge from X, the system would have been able to link the contract, Bob is about to interact with, to all of Homer_eth's prior blockchain activity. The KG would have revealed a critical anomaly. Instead of the mint proceeds being transferred to the contract address of the project for future development, they were being diverted to the *Ether Reapers* deployer address via the *MintReaper* function. With smart contracts acting as an automated and trustless intermediary, where the code of the contract dictates the flow of funds according to predefined rules, this pattern of fund diversion is absent in legitimate projects. When funds are sent directly to a team member's address, in this case the deployer address, the funds can be moved to exchanges or other addresses with ease (i. e., pulling liquidity from the project without fulfilling the promises). This is a common tactic in rug pulls, where the developers abandon the project and disappear with the investor funds. Therefore, signaling a potential rug pull behavior. Upon detecting this anomaly, the system would have immediately issued a rug pull warning to Bob, prompting Bob to make an informed decision by asking whether he wishes to proceed with the transaction despite the identified risk. This proactive approach empowers Bob to reconsider his decision with full awareness of the potential danger, offering him a chance to opt-out before potentially falling victim to a rug pull.

6 Future Work

In future work, we aim to translate the Kosmosis approach into a practical implementation. This will involve developing a pipeline architecture that supports the integration of incremental changes without necessitating the reconstruction of the entire knowledge graph. The incremental changes are append-only, without applying updates or deletes. Testing and evaluation will form another critical component of this future work. The implementation will be validated in the form of performance metrics, including update latency, graph completeness, and scalability. Synthetic datasets and real-world data streams will be utilized to simulate diverse and challenging scenarios, ensuring the generalizability and robustness of the approach, e. g. that the matching between different data sources is reliable. Furthermore, an alerting system that utilizes the KG, constructed with Kosmosis, to alert users before interacting with a potential rug pull project, as described in the user story of Section 5, requires future efforts.

7 Conclusion

The Kosmosis approach represents a significant advancement in addressing the challenges associated with crypto rug pulls. Our proposed approach offers enhanced capabilities for semantic analysis, allowing the identification of fraud patterns that traditional transaction graph methods cannot detect.

We outlined a user story, where a threat actor known as Homer_eth executed five NFT project heists within two months, accumulating over $2.8 million in profits. In such scenario, we showed that Kosmosis provides a knowledge graph that improves the detection of such fraudulent schemes carried out through sophisticated transaction patterns that might otherwise go unnoticed in related approaches, such as smart contract code analysis. This capability helps users make informed decisions and avoid becoming victims of fraud.

The Kosmosis approach supports the ingestion of unstructured, semi-structured, and structured data, as well as the ingestion of new data at different time intervals. During construction, the semantics of blockchain transactions are extracted to address “why” and “how” crypto assets were transferred. Thus, Kosmosis extends the traditional transaction graph into a *semantically enhanced transaction graph* in which the sender and recipient are still pseudonyms. By incrementally constructing a *knowledge graph from blockchain and social media data*, Kosmosis also bridges the gap between pseudonymous transactions and real-world entities.

References

[Al23] Alizadeh, S.; Setayesh, A.; Mohamadpour, A.; Bahrak, B.: A network analysis of the non-fungible token (NFT) market: structural characteristics, evolution, and interactions. Applied Network Science 8 (1), p. 38, 2023.

[Ba22] Bauer, T.; Beer, F.; Holl, D.; Imeraj, A.; Schweiger, K.; Stangl, P.; Weigl, W.; Neumann, C. P.: Reddiment: Eine SvelteKit- und ElasticSearch-basierte Reddit Sentiment-Analyse, tech. rep. CL-2022-06, Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, 2022, DOI: 10.13140/RG.2.2.32244.12161.

[Bé21] Béres, F.; Seres, I. A.; Benczúr, A. A.; Quintyne-Collins, M.: Blockchain is Watching You: Profiling and Deanonymizing Ethereum Users. In: 2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS). IEEE, Online, pp. 69–78, 2021, DOI: 10.1109/DAPPS52256.2021.00013.

[Ch20a] Chakravarty, M. M.; Chapman, J.; MacKenzie, K.; Melkonian, O.; Peyton Jones, M.; Wadler, P.: The extended UTXO model. In: Financial Cryptography and Data Security: FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC, Kota Kinabalu, Malaysia, February 14, 2020, Revised Selected Papers 24. Springer, pp. 525–539, 2020.

[Ch20b] Chen, L.; Peng, J.; Liu, Y.; Li, J.; Xie, F.; Zheng, Z.: Phishing Scams Detection in Ethereum Transaction Network. ACM Trans. Internet Technol. 21 (1), 2020, ISSN: 1533-5399, DOI: 10.1145/3398071, URL: https://doi.org/10.1145/3398071.

[Ch23a] Chainalysis: The 2023 Crypto Crime Report, Chainalysis, 2023, URL: https://go.chainalysis.com/2023-crypto-crime-report.html.

[Ch23b] Chernysheva, A.; Götz, J.; Imeraj, A.; Korinth, P.; Stangl, P.; Neumann, C. P.: SGDb Semantic Video Game Database: Svelte- und Ontotext-basierte Webanwendung mit einer Graphen-Suche für Videospiele, tech. rep. CL-2023-02, Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, 2023, DOI: 10.13140/RG.2.2.11272.60160.

[EL23] Elmougy, Y.; Liu, L.: Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '23, Association for Computing Machinery, Long Beach, CA, USA, pp. 3979–3990, 2023, DOI: 10.1145/3580305.3599803, URL: https://doi.org/10.1145/3580305.3599803.

[En18] Entriken, W.; Shirley, D.; Evans, J.; Sachs, N.: ERC-721: Non-Fungible Token Standard, Ethereum, 2018, URL: https://eips.ethereum.org/EIPS/eip-721.

[FW16] Feilmayr, C.; Wöß, W.: An analysis of ontologies and their success factors for application to business. Data & Knowledge Engineering 101, pp. 1–23, 2016.

[Ha22a] Hahn, B.; Kleber, M.; Klier, A.; Kreussel, L.; Paris, F.; Ziegler, A.; Neumann, C. P.: TwitterDash: React- und .NET-basierte Trend- und Sentiment-Analysen, tech. rep. CL-2022-07, Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, 2022, DOI: 10.13140/RG.2.2.15466.90564.

[Ha22b] Halbritter, J.; Kohl, H.; Kreussel, L.; Prettner, S.; Ziegler, A.; Neumann, C. P.: Graphvio: Eine Graphdatenbank-Webanwendung für integrierte Datensätze von Streaminganbietern, tech. rep. CL-2022-01, Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, 2022, DOI: 10.13140/RG.2.2.12111.46244.

[Ho17] Hoskinson, C.: Why we are building Cardano, IOHK, 2017, URL: https://api-new.whitepaper.io/documents/pdf?id=HkUIhFWhL.

[Ho22] Hogan, A.; Blomqvist, E.; Cochez, M.; Gutierrez, C.; Gayo, J. E. L.; Kirrane, S.; Neumaier, S.; Polleres, A.; Navigli, R.; Ngomo, A.-C. N.; Rashid, S. M.; Rula, A.; Schmelzeisen, L.; Sequeda, J.; Staab, S.; Zimmermann, A.: Knowledge Graphs. ACM Computing Surveys 54 (4), pp. 1–37, 2022, ISSN: 0360-0300, 1557-7341, DOI: 10.1145/3447772, arXiv: 2003.02320 [cs.AI], URL: http://arxiv.org/abs/2003.02320, visited on: 12/11/2023.

[Ho23] Hofer, M.; Obraczka, D.; Saeedi, A.; Köpcke, H.; Rahm, E.: Construction of Knowledge Graphs: State and Challenges, 2023, eprint: 2302.11509 (cs.AI).

[Hu21] Huang, H.; Kong, W.; Zhou, S.; Zheng, Z.; Guo, S.: A survey of state-of-the-art on blockchains: Theories, modelings, and tools. ACM Computing Surveys (CSUR) 54 (2), pp. 1–42, 2021.

[Kh22] Khan, A.: Graph analysis of the ethereum blockchain data: A survey of datasets, methods, and future work. In: 2022 IEEE International Conference on Blockchain (Blockchain). IEEE, IEEE, Espoo, Finland, pp. 250–257, 2022.

[Ma23] Marin, O.; Cioara, T.; Toderean, L.; Mitrea, D.; Anghel, I.: Review of Blockchain Tokens Creation and Valuation. Future Internet 15 (12), p. 382, 2023, ISSN: 1999-5903, DOI: 10.3390/fi15120382, URL: https://www.mdpi.com/1999-5903/15/12/382, visited on: 12/17/2023.

[MAD22] Mazorra, B.; Adan, V.; Daza, V.: Do not rug on me: Zero-dimensional Scam Detection, 2022, eprint: 2201.07220 (cs.CR).

[Me23] MetaMask: MetaMask: A crypto wallet & gateway to blockchain apps, ConsenSys Software Inc., 2023, URL: https://metamask.io/.

[Mo23] Moody, R.: Worldwide crypto & NFT rug pulls and scams tracker, Comparitech, 2023, URL: https://www.comparitech.com/crypto/cryptocurrency-scams/.

[Na08] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, 2008, URL: https://bitcoin.org/bitcoin.pdf.

[NFL10] Neumann, C. P.; Fischer, T.; Lenz, R.: OXDBS – Extension of a native XML Database System with Validation by Consistency Checking of OWL-DL Ontologies. In: Proc of the 14th International Database Engineering & Applications Symposium (IDEAS'10). Montreal, QC, CA, pp. 143–148, 2010, DOI: 10.1145/1866480.1866502.

[SAS23] Sharma, T.; Agarwal, R.; Shukla, S. K.: Understanding Rug Pulls: An In-Depth Behavioral Analysis of Fraudulent NFT Creators. ACM Trans. Web 18 (1), 2023, ISSN: 1559-1131, DOI: 10.1145/3623376, URL: https://doi.org/10.1145/3623376.

[SN23] Stangl, P.; Neumann, C. P.: FoodFresh: Multi-Chain Design for an Inter-Institutional Food Supply Chain Network. In: Proc of the 14th International Conference on Cloud Computing, GRIDs, and Virtualization (Cloud Computing 2023). Nice, France, pp. 41–46, 2023, DOI: 10.48550/ARXIV.2310.19461.

[St22] Stangl, P.: Design and Implementation of a Heterogeneous Blockchain Consortium for a Food Supply Chain Network, Bachelor's Thesis, Ostbayerische Technische Hochschule Amberg-Weiden, 2022, URL: https://www.cyberlytics.eu/theses/all/OTH-AW/BT_2022_Stangl_Philipp_Thesis/BT_2022_Stangl_Philipp_Thesis.pdf.

[VB15] Vogelsteller, F.; Buterin, V.: ERC-20: Token Standard, Ethereum, 2015, URL: https://eips.ethereum.org/EIPS/eip-20.

[Wa19] Wang, S.; Li, D.; Zhang, Y.; Chen, J.: Smart contract-based product traceability system in the supply chain scenario. IEEE Access 7, pp. 115122–115133, 2019.

[Wo24] Wood, G.: Ethereum: A Secure Decentralised Generalised Transaction Ledger, (Ethereum project yellow paper), Parity Technologies, 2024, URL: https://ethereum.github.io/yellowpaper/paper.pdf.

[Za22] ZachXBT [@zachxbt]: Homer.eth (formerly @homer_eth) Rug Pull Analysis, X, X Corp., 2022, URL: https://x.com/zachxbt/status/1529973318563946496.

[Zh17] Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H.: An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In: 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, Boston, MA, USA, pp. 557–564, 2017, DOI: 10.1109/BigDataCongress.2017.85.

[Zh21] Zhu, X.; Ao, X.; Qin, Z.; Chang, Y.; Liu, Y.; He, Q.; Li, J.: Intelligent financial fraud detection practices in post-pandemic era. The Innovation 2 (4), 2021.

[Zh24] Zhou, Y.; Sun, J.; Ma, F.; Chen, Y.; Yan, Z.; Jiang, Y.: Stop Pulling my Rug: Exposing Rug Pull Risks in Crypto Token to Investors. 2024.

Self-Refinement Strategies for LLM-based Product Attribute Value Extraction

Alexander Brinkmann [1] and Christian Bizer [1]

Abstract: Structured product data, in the form of attribute-value pairs, is essential for e-commerce platforms to support features such as faceted product search and attribute-based product comparison. However, vendors often provide unstructured product descriptions, making attribute value extraction necessary to ensure data consistency and usability. Large language models (LLMs) have demonstrated their potential for product attribute value extraction in few-shot scenarios. Recent research has shown that self-refinement techniques can improve the performance of LLMs on tasks such as code generation and text-to-SQL translation. For other tasks, the application of these techniques has resulted in increased costs due to processing additional tokens, without achieving any improvement in performance. This paper investigates applying two self-refinement techniques — error-based prompt rewriting and self-correction — to the product attribute value extraction task. The self-refinement techniques are evaluated across zero-shot, few-shot in-context learning, and fine-tuning scenarios using GPT-4o. The experiments show that both self-refinement techniques fail to significantly improve the extraction performance while substantially increasing processing costs. For scenarios with development data, fine-tuning yields the highest performance, while the ramp-up costs of fine-tuning are balanced out as the amount of product descriptions increases.

Keywords: Self-Refinement, Information Extraction, Large Language Models, E-Commerce

1 Introduction

Large Language Models (LLMs), such as OpenAI's GPT-4o, have been successfully applied to a wide range of tasks, including information extraction tasks such as extracting product attribute values from product descriptions [BSB25]. In order to be effective on these tasks, LLMs often rely on few-shot in-context learning and fine-tuning. Recently, methods for the automated self-refinement of prompts [Ma23] and the self-review and self-correction of model decisions [Pa24] have emerged and are successfully applied for tasks such as code generation [Ma23] and text-to-SQL translation [PR23]. At the same time, an increasing body of research [Ol23, Hu23] criticises the self-refinement approaches as for other tasks, they do not significantly improve performance while increasing the processing costs heavily due to additional tokens that need to be processed. This paper critically evaluates two self-refinement techniques for extracting attribute values from product descriptions: error-based prompt rewriting and self-correction. Error-based prompt rewriting improves the attribute definitions within prompts by analyzing errors made by the model on validation examples. Self-correction reviews and updates the initial output of an LLM if it spots wrongly

[1] University of Mannheim, Data & Web Science Group, Schloss, 68161 Mannheim, Germany,
alexander.brinkmann@uni-mannheim.de, https://orcid.org/0000-0002-9379-2048;
christian.bizer@uni-mannheim.de, https://orcid.org/0000-0003-2367-0237

doi:10.18420/BTW2025-132

extracted values. These two self-refinement techniques are chosen as they can be applied in a fully automated fashion and cover self-refinement during prompt engineering (error-based prompt rewriting) and post hoc self-refinement of the LLM's output (self-correction). The self-refinement techniques are evaluated in zero-shot, few-shot in-context learning and fine-tuning scenarios. This paper makes the following contributions:

- The self-refinement techniques error-based prompt rewriting and self-correction are experimentally evaluated for the product attribute value extraction task. The self-refinement techniques are applied in zero-shot, few-shot in-context learning, and fine-tuning scenarios using GPT-4o.
- We present a detailed analysis of the impact of error-based prompt rewriting on prompt quality and the effect of self-correction on the accuracy of the extracted attribute values.

The paper is structured as follows. First, related work is reviewed. Section 3 and Section 4 describe the datasets used and the experimental setup. Section 5 introduces product attribute value extraction using few-shot learning and fine-tuning. The self-refinement strategies error-based prompt rewriting and self-correction are introduced and experimentally evaluated in Section 6 and Section 7. Code and data for replicating all experiments are available online[2].

2 Related Work

Attribute Value Extraction. Product attribute value extraction is a subtask of information extraction and focuses on extracting specific attribute values from unstructured text, such as product titles and descriptions [ZZH09, Ya22]. There are two variants of the task: closed-world, where a predefined schema specifies the target attributes, and open-world, where the set of target attributes is undefined and the extraction method needs to determine the attributes as well as their values [Zh22, Xu24]. We focus on closed-world product attribute value extraction. Early works use domain-specific rules to identify attribute values in product descriptions [ZZH09]. Recently, many approaches have framed product attribute value extraction as a question-answering task, using the pre-trained language model (PLM) BERT to identify the target attribute value in a product description [Ya22]. Other related works use LLMs like GPT-4 to extract attribute values from product descriptions using different prompting techniques [Fa24, BSB25].

Information Extraction using LLMs. Generative LLMs often demonstrate superior zero-shot performance compared to PLMs and exhibit higher robustness for unseen examples [Br20]. This advantage is due to extensive pre-training on large amounts of text and emergent abilities arising with large model size [We22]. LLMs have been successfully applied to information extraction tasks across various domains [Xu24]. For instance, Wang et al.[WLJ23] and Parekh et al.[Pa23] utilized OpenAI's LLMs to extract event data from

[2] https://github.com/wbsg-uni-mannheim/SelfRefinement4ExtractGPT

unstructured text. Goel et al. [Go23] combined LLMs with human expertise to annotate patient information in medical texts.

Self-Refinement Techniques. Various techniques for model self-refinement and for correcting model outputs have been proposed recently [Pa24]. [Ma23] used automated feedback generated by an LLM to improve the readability of code. The *Self-Correction* technique evaluated in this paper also relies on such automated feedback. A critic of self-correction is that LLMs struggle to correct their responses without external feedback from tools such as code interpreters or database management systems [Hu23]. An example of how LLMs reflect on the feedback from tools to improve code successfully is given in [Sh23], while [Ol23] finds GPT-4 that is not able to generate useful feedback for fixing mistakes in code. The *Error-based Prompt Rewriting* technique that is evaluated in this paper tries to improve the initial prompt using development data as a source of external feedback.

3 Datasets

This section introduces the benchmark datasets OA-Mine [Zh22] and AE-110k [Xu19] that we use for the experiments. Both datasets consist of English product offers with annotated attribute-value pairs and have been used in related work [BSB25, Ya22].

OA-Mine. We use a subset of the human-annotated product offers of the OA-Mine dataset[3] [Zh22] for our experiments. The subset includes 10 product categories, with up to 80 product offers per category. Each category has between 8 to 15 attributes, resulting in a total of 115 unique attributes. Attributes with the same name in different product categories are treated as distinct attributes. We do not apply any further pre-processing to the offers.

AE-110K. The AE-110K dataset[4] comprises triples of product titles, attributes and attribute values from the AliExpress Sports & Entertainment category [Xu19]. Product offers are derived by grouping the triples by product title. The subset includes 10 product categories, with up to 160 product offers per category. For each category, 6 to 17 attributes are known, resulting in a total of 101 unique attributes.

Development/Test Split. Table 1 contains statistics about the numbers of unique attribute-value pairs (A/V pairs), unique attribute values, and product offers for all four datasets. We use the same 60:40 development and test split based on product offers as in related work [BSB25]. The examples in the development set are labelled with ground truth attribute-value pairs. We use the development set to sample example attribute values, to select in-context demonstrations, as a validation set for error-based prompt rewriting, and as a training set for fine-tuning.

[3] https://github.com/xinyangz/OAMine/tree/main/data

[4] https://raw.githubusercontent.com/lanmanok/ACL19_Scaling_Up_Open_Tagging/master/publish_data.txt

Tab. 1: Descriptive statistics for OA-Mine and AE-110K.

	OA-Mine			AE-110K		
	Development	Test	Total	Development	Test	Total
A/V Pairs	3,626	2,451	6,077	2,170	1,482	3,652
Unique A/Vs	2,400	1,749	3,637	587	454	854
Product Offers	715	491	1,206	785	524	1,309

Example Extractions. Table 2 shows example product offer titles, target attributes and attribute values from the datasets. The examples (a) and (b) visualize the direct extraction meaning that the extracted attribute value is a substring of the product title. The target output is a JSON object containing all attribute-value pairs.

Tab. 2: Example product titles and attribute-value pairs from the OA-Mine and AE-110k datasets.

Dataset	**OA-Mine**	**AE-110k**
Category Attributes	Vitamin Brand, Net Content, Supplement Type, Dosage	Eyewear Sport Type, Gender Lenses Optical, Model Number
Product Title	NOW Supplements, Vitamin A (Fish Liver Oil) 25,000 IU, Essential Nutrition, 250 Softgels	Professional Men Ski Goggles Double Layers UV400 Anti-fog Big Ski Mask Skiing Snowboard Glasses
Target Output	´´´´json{ “Brand“: “Now Supplements“, “Net Content“: “250“, “Supplement Type“: “Vitamin A“, “Dosage“: “25,000 IU“ }´´´´	´´´´json{ “Sport Type“: “Skiing“, “Gender“: “Men“, “Lenses Optical“: “UV400“, “Model Number“: “n/a“ }´´´´
	(a)	(b)

4 Experimental Setup

We use the LLM gpt-4o-2024-08-06[5] for all experiments in this paper. GPT-4o is chosen because, at the time of experimenting for this paper, it is the LLM with the best performance on benchmarks measuring natural language understanding[6]. GPT-4o is accessed via the OpenAI API. GPT-4o's temperature parameter is set to zero to reduce randomness. We report average F1-scores of three runs for each experiment and run paired t-tests to verify with 99% confidence if the F1-scores of two experiments are significantly different. The F1-score is calculated by categorizing predictions into five groups as per previous works [Xu19, Ya22, BSB25, BBB24]. The five categories are NN (no predicted value,

[5] https://platform.openai.com/docs/models/gp#gpt-4o
[6] https://openai.com/index/hello-gpt-4o/

no ground truth value), NV (predicted value, no ground truth value), VN (no predicted value, ground truth value), VC (predicted value exactly matches ground truth value), and VW (predicted value does not match ground truth value). The F1-score is derived from precision ($P = VC/(NV+VC+VW)$), recall ($R = VC/(VN+VC+VW)$), and the formula $F1 = 2PR/(P+R)$. Additionally, we report the average number of tokens per prompt to estimate and compare the costs of the different approaches. It is important to mention that prompt and completion tokens are summed. OpenAI charges users different prices for input tokens (2.5$/1M tokens) and output tokens (1.25$/1M tokens)[7].

5 Attribute Value Extraction using In-Context Learning and Fine-Tuning

This section introduces LLM-based prompting techniques for attribute value extraction. It covers zero-shot prompts, prompts with attribute definitions generated from development data, few-shot in-context learning prompts, few-shot in-context learning prompts combined with self-consistency and fine-tuning on the development sets.

Zero-Shot. Figure 1 shows the zero-shot prompt consisting of a task description and a task input. The task description is a system chat message that describes the attribute value extraction task, lists the target attributes, defines the output as a JSON object, and explains that not available attribute values should be marked with `'n/a'` in the output JSON object. The task input is a user chat message that contains the product description from which the attribute values are extracted. Examples of output JSON objects are shown in Table 2.

Task Description (System)	Extract the attribute values for **'Brand', 'Supplement Type', 'Dosage', 'Net Content'** into a JSON object. Return 'n/a' if the attribute is not found.
Task Input (User)	NOW Supplements, Vitamin A (Fish Liver Oil) 25,000 IU, Essential Nutrition, 250 Softgels

Fig. 1: Zero-shot prompt.

Row *Zero-Shot* in Table 3 shows the results for zero-shot attribute value extraction experiments. The F1-scores remain below 70%, which is unsatisfying for real-world deployment. The standard deviation of the reported average F1-scores is 0.11 and 0.45 for OA-Mine and AE-110k, showing that the extracted attribute values only marginally deviate across the runs. We use the results of the zero-shot experiment as a reference point for the upcoming experiments. We report the delta between each experimental F1-score and the zero-shot F1-score as Δ ZS. We report the ratio of the number of tokens to the number of tokens used by the zero-shot prompt as token factor (TF).

Attribute Definitions. In the attribute definitions scenario, attribute definitions are appended to the task description of the zero-shot prompt. This prompt builds on the findings of

[7] OpenAI prices as of February 2025: https://openai.com/api/pricing/

Tab. 3: Experimental results for attribute value extraction using in-context learning and fine-tuning.

	OA-Mine				AE-110k			
	F1	**Δ ZS**	**Tokens**	**TF**	**F1**	**Δ ZS**	**Tokens**	**TF**
Zero-shot	68.8	0.0	181	1.0	63.6	0.0	196	1.0
Attribute Definitions	72.2	+3.4	601	3.3	76.3	+12.7	577	2.9
Few-Shot	78.6	+9.8	1,315	7.2	83.9	+20.3	1,351	6.9
+ Self-Consistency	79.3	+10.5	3,945	21.7	84.1	+20.4	4,053	20.6
+ Attribute Definitions	79.3	+10.5	1,760	9.7	85.3	+21.7	1,727	8.8
Fine-Tuning	83.2	+14.3	172	0.9	85.1	+21.4	177	0.9

previous research demonstrating that definitions enhance the performance of LLMs for classification [Pe23] and extraction tasks [BSB25]. The example prompt in Figure 2 depicts this extension. For demonstration purposes, it contains only the definition for the attribute `'Brand'`. In the experiments, a definition is appended for each attribute. Since the datasets OA-Mine and AE-110k do not contain attribute definitions, the definitions are generated by an LLM based on five attribute values. The attribute values are randomly sampled from the development set. Generating attribute definitions requires an average of 84 and 74 tokens per unique attribute for OA-Mine and AE-110k, respectively. The generated attribute definitions have an average length of 36 tokens for OA-Mine and 35 tokens for AE-110k.

Task Description (System)	Extract the attribute values for 'Brand', 'Supplement Type', 'Dosage', 'Net Content' into a JSON object. Return 'n/a' if the attribute is not found. **Attribute Definitions:** **Brand**: The 'Brand' attribute refers to the name of the company that produces the product. Examples include 'Pure Synergy', and 'Best Naturals'.
Task Input (User)	NOW Supplements, Vitamin A (Fish Liver Oil) 25,000 IU, Essential Nutrition, 250 Softgels

Fig. 2: Prompt with attribute definitions.

Table 3 shows that compared to the zero-shot results, the attribute definitions improve GPT-4o's F1-score by 3% for OA-Mine and nearly 13% for AE-110k, which are significant differences (verified using paired t-tests with 99% confidence), but the prompts including the definitions are approximately 3 times as long as the zero-shot prompts (TF=3.3 and TF=2.9). This result underlines the usefulness of attribute definitions for extracting product attribute values.

Few-Shot In-Context Learning. In the few-shot scenario, demonstrations with annotated attribute-value pairs are presumably available as a development set. For few-shot learning, the zero-shot prompt and the prompt with attribute definitions are extended with demonstrations. Figure 3 shows the extension of the zero-shot prompt. Each demonstration consists of a

demonstration task input and a demonstration task output. The demonstration task input is a user message containing a product description. The demonstration task output is an assistant message with the extracted attribute-value pairs formatted as a JSON object. We use demonstrations that are semantically similar to the current product offer. For selecting these demonstrations, the demonstrations of the development set are embedded using OpenAI's embedding model `text-embedding-ada-002`[8]. The embedded demonstrations with the greatest cosine similarity to the embedded task input are considered to be semantically similar. Following [BSB25], we add 10 demonstrations to each prompt.

Task Description (System)	Extract the attribute values for 'Brand', 'Supplement Type', 'Dosage', 'Net Content' into a JSON object. Return 'n/a' if the attribute is not found.
Demonstration – Task Input (User)	NOW Supplements, Vitamin C Crystals (Ascorbic Acid), Antioxidant Protection*, 1-Pound
Demonstration –Task Output (Assistent)	´´´´json { "**Brand**": "NOW Supplements", "**Supplement type**": "Ascorbic Acid", "***Dosage***": "n/a", "**Net content**": "1-Pound" } ´´´´
Task Input (User)	NOW Supplements, Vitamin A (Fish Liver Oil) 25,000 IU, Essential Nutrition, 250 Softgels

Fig. 3: Few-shot prompt.

Row *Few-Shot* in Table 3 reports the results of the few-shot experiments. As expected, few-shot in-context learning with semantically similar demonstrations significantly improves GPT-4o's F1-score but more than doubles the number of tokens per prompt compared to adding attribute definitions (e.g. the TF for OA-Mine rises from 3.3 to 7.2). The gains show that GPT-4o requires demonstrations to achieve F1-scores above or at least close to 80%. Attribute definitions increase the F1-score by 1%, which is a significant increase according to the t-tests, but adds around 400 tokens to the prompt (TF=9.7 and TF=8.8).

Few-Shot Self-Consistency. Self-consistency extends the few-shot in-context learning prompt by sampling three outputs from the LLM and selecting the most consistent answer through majority voting for each attribute. Each run attribute order in the prompt is shuffled to avoid an ordering bias [ZYW25]. Related work proposes self-consistency as an alternative to self-refinement with similar token usage [Hu23]. Row *Self-consistency* in Table 3 reports the experimental results. Due to the three runs, few-shot in-context learning with self-consistency costs three times the number of tokens compared to few-shot in-context learning. The ensemble runs have a low deviation in the extracted attribute values explaining the insignificant performance changes.

Fine-Tuning. In-context learning provides task-specific knowledge to the LLM via the prompt. Fine-tuning uses training data to encode task-specific knowledge into the parameters of the LLM. At runtime, this task-specific knowledge is implicitly used to extract attribute

[8] https://platform.openai.com/docs/guides/embeddings/

values. In preparation for fine-tuning, the development records are formatted with the zero-shot prompt consisting of a task description, an input and an output containing attribute-value pairs. To execute the fine-tuning, the pre-processed datasets are uploaded to OpenAI's fine-tuning API[9] and GPT-4o is trained for three epochs on the uploaded datasets using OpenAI's default parameters. Row *Fine-Tuning* in Table 3 reports the fine-tuning results for GPT-4o. The fine-tuned GPT-4o LLMs achieve the highest average F1-score with the lowest token usage during application (TF=0.9). But especially on the dataset AE-110k, few-shot in-context learning is a competitive alternative to fine-tuning. The initial investment required for fine-tuning is 394k and 431k tokens, respectively, for OA-Mine and AE-110k. Based on the number of tokens and the token cost per product offer, a break-even point concerning the number of test product offers for using fine-tuning GPT-4o instead of few-shot in-context learning can be calculated as follows: $FineTuningCost/(TokenCost_{FewShot} - TokenCost_{FineTuning})$. Using this formula the average break-even point for the datasets is 6,666 product offers. For scenarios in which attribute values have to be extracted from more than 6,666 product offers, it is cheaper to invest in fine-tuning than to use the base model with the longer in-context learning prompts.

6 Error-based Prompt Rewriting

This section introduces and evaluates the self-refinement technique of error-based prompt rewriting. The method is combined with the zero-shot and few-shot in-context learning prompts that use attribute definitions introduced in Section 5.

Prompting Technique. Error-based prompt rewriting uses the development set to improve the attribute definitions in the prompts. It assumes that better attribute definitions improve the product attribute value extraction [Pe23]. For error-based prompt rewriting, the prompts with attribute definitions are run on five randomly selected product offers from the development set for each category to extract attribute values. The extracted attribute values are compared to the ground truth from the development set to identify incorrectly extracted values. For each attribute with extraction errors, the prompt shown in Figure 4 is populated with the existing attribute definition and a list of product descriptions with incorrectly and correctly extracted attribute values. In the depicted example, '60' instead of '60 Capsules' is expected. The LLM responds to the prompt with a rewritten attribute definition. Table 4 illustrates how the attribute definition for the `Net Content` is rewritten to be more specific. Error-based prompt rewriting is repeated three times to evaluate the product attribute value extraction on up to 15 product descriptions to improve the attribute definitions.

Discussion of Results. Rewriting attribute definitions increases token usage. In the zero-shot scenario, 270k tokens are required on OA-Mine for rewriting attribute definitions, while 350k tokens are consumed on AE-110k. These values rise to 378k and 389k tokens in the few-shot scenario. Table 5 reports the experimental results of the product attribute

[9] https://platform.openai.com/docs/guides/fine-tuning

Task Description (System)	You are provided with an attribute definition that is used to extract attribute values from product descriptions and a list of product descriptions with incorrectly and correctly extract attribute values. Improve the attribute definition based on the incorrect and correct attribute values. Attribute values should be mentioned in "" and should naturally occur in the attribute definition. Respond with JSON object in the format: attribute: attribute definition with example values.
Task Input (User)	**Type**: The 'Net content' attribute indicates the amount of product contained in the package, which can be expressed in various units like '4 oz', '500', or '1 lb (454 Grams)'. **Product Offer**: Thorne Research - Quercenase - Quercetin Phytosome Supplement with Bromelain - 60 Capsules **Incorrect value**: 60 Capsules **Correct value**: 60

Fig. 4: Prompt for error-based rewriting of attribute definitions.

Tab. 4: Definitions for the attribute `'Net Content'` of a vitamin product.

Extraction Error	Attribute Definition	Rewritten Attribute Definition
Extracted Value: "60 Capsules" Expected Value: "60"	The 'Net Content' attribute indicates the amount of product contained in the package, which can be expressed in various units like '4 oz', '500', or '1 lb (454 Grams)'.	The 'Net Content' attribute specifies the quantity of product within a package. For products like capsules, or softgels, it is expressed as a numerical value, such as '60'. For liquid products, it is expressed in fluid ounces, such as '0.08 Fl Oz'. For products sold by weight, it should be expressed in pounds or grams, such as '1 lb (454 Grams)'.

extraction without and with rewritten attribute definitions. In the zero-shot scenario, the rewritten definitions insignificantly degrade GPT-4o's performance (verified by t-tests with 99% confidence) and cause an increased token usage (e.g. the TF for OA-Mine rises from 3.3 to 8.6). In the few-shot learning configuration, the performance does not significantly change, though token usage rises again (e.g. the TF for OA-Mine rises from 9.7 to 13.9).

Analysis of Rewritten Attribute Definitions. The analysis of error-based attribute definition rewriting focuses on the OA-Mine dataset in the zero-shot scenario. To reduce the manual effort of analyzing all changes, GPT-4o is used to analyze the attribute definition. GPT-4o analyzes the length of the attribute definitions, the level of detail of the original and the rewritten attribute definition, counts the number of example values per attribute definition and checks if values are excluded in an attribute definition. We manually review 10 analyzed rewriting operations to estimate how well GPT-4o performs this analysis. In the manual assessment, the level of detail and the mention of excluded example values are always correct. The example value counts are in 80% of the cases correct. With a tolerance range of

Tab. 5: Experimental results for error-based prompt rewriting.

	OA-Mine				AE-110k			
	F1	**Δ ZS**	**Tokens**	**TF**	**F1**	**Δ ZS**	**Tokens**	**TF**
Attribute Definitions	72.2	+3.4	601	3.3	76.3	+12.7	577	2.9
+ Rewriting	71.8	+2.9	1,569	8.6	74.4	+10.8	1,872	9.5
Few Shot + Attr. Defs.	79.3	+10.5	1,760	9.7	85.3	+21.7	1,727	8.8
+ Rewriting	79.0	+10.2	2,520	13.9	85.5	+21.8	2,650	13.5

two, all example value counts are valid. Hence, GPT-4o is useful for this analysis. Over three iterations, GPT-4o completed 176 out of 345 possible rewrites. The 345 possible rewrites are calculated by multiplying the number of unique attributes (115) in OA-mine by three because every iteration each unique attribute definition can but must not be rewritten once. The quantitative analysis of the attribute definitions shows that 97% of the rewritten attribute definitions are longer than the original attribute definition. The longer attribute definitions explain the higher TF reported in Table 5. 93% of GPT-4o's rewriting operations enhance the level of detail. In 65% of the operations, the level of detail is increased by adding example values and in 31% of the operations, the level of detail is increased by explicitly excluding attribute values. The enhanced level of detail through additional information and the inclusion and exclusion of attribute values leads to an overfitting of the attribute definitions to the development set, which harms the LLM's performance on the test set.

7 Self-Correction

This section introduces and evaluates the post hoc self-refinement technique self-correction. Self-correction can be combined with all prompts introduced in Section 5.

Prompting Technique. Motivated by related work [Ma23], the LLM post hoc reviews and updates its initially extracted attribute values. Therefore, a first prompt instructs the LLM to extract attribute-value pairs from the input. This first prompt is one of the prompts introduced in Section 5 depending on the scenario (zero-shot, few-shot in-context learning or fine-tuning). The output of the first prompt is sent to the same LLM again with a request to reflect on and correct erroneously extracted attribute values. Figure 5 illustrates the second self-correction prompt. A criticism of related work on the original self-refinement paper is that the refinement prompt provides additional task-related information [Hu23]. To ensure that similar information is provided by the initial prompt and the self-correction prompt, attribute definitions and few-shot in-context learning demonstrations are added to the prompt if the initial prompt contains them. In the fine-tuning scenario, the fine-tuned GPT-4o executes the self-correction prompt.

Task Description (System)	Check if the attribute values for the attributes 'Brand', 'Supplement Type', 'Dosage', 'Net Content' are correctly extracted. Respond with a JSON object that contains correctly extracted attribute values. Return 'n/a' if the attribute is not found.
Task Input (User)	NOW Supplements, Vitamin A (Fish Liver Oil) 25,000 IU, Essential Nutrition, 250 Softgels **Extracted Attribute Values**: { "**Brand**": "NOW Supplements", "**Supplement type**": "Vitamin A", "**Dosage**": "25,000 IU", "**Net content**": "n/a" }

Fig. 5: Prompt for self-correction.

Discussion of Results. Table 6 shows that in the zero-shot scenario, the performance gains from self-correction are insignificant (verified by t-tests with 99% confidence) and do not justify the substantial increase in computational cost (TF=2.6). In the few-shot scenario with in-context learning demonstrations, self-correction does not significantly alter GPT-4o's performance while more than doubling token usage (e.g. the TF for OA-Mine rises from 7.2 to 15.2). In the fine-tuning scenario, the performance differences are insignificant but again the token usage doubles (e.g. the TF for OA-Mine rises from 0.9 to 2.5). These findings indicate that self-correction is not cost-effective and only marginally influences the performance of GPT-4o for attribute value extraction across different learning scenarios.

Tab. 6: Experimental results for self-correction.

	OA-Mine				**AE-110k**			
	F1	**Δ ZS**	**Tokens**	**TF**	**F1**	**Δ ZS**	**Tokens**	**TF**
Zero-shot	68.8	0.0	181	1.0	63.6	0.0	196	1.0
+ Self-Correction	69.3	+0.4	467	2.6	63.8	+0.1	503	2.6
Attribute Definitions	72.2	+3.4	601	3.3	76.3	+12.7	577	2.9
+ Self-Correction	72.5	+3.6	1,305	7.2	77.2	+13.5	1,285	6.5
Few-Shot	78.6	+9.8	1,315	7.2	83.9	+20.3	1,351	6.9
+ Self-Correction	78.5	+9.6	2,751	15.2	83.7	+20.0	2,815	14.3
+ Attribute Definitions	79.3	+10.5	1,760	9.7	85.3	+21.7	1,727	8.8
+ Attr. Def. & Self-Corr.	78.7	+9.8	3,589	19.8	84.9	+21.3	3,585	18.3
Fine-Tuning	83.2	+14.3	172	0.9	85.1	+21.4	177	0.9
+ Self-Correction	82.9	+14.1	449	2.5	85.2	+21.6	457	2.3

Analysis of Self-Corrected Attribute Values. We analyze the self-correction in detail by distinguishing three possible outcomes of the self-correction step: (1) a wrong value is corrected (Improvement), (2) a previously correct value is corrupted (Corruption), and (3) the update of the extracted value does not correct a wrong value but just changes it (Still wrong). Table 7 provides examples of the three outcomes. The target attribute value is underlined in the product description. A quantitative analysis of the zero-shot scenario

shows that 64% of the 165 attribute value updates on OA-Mine and 90% of the 423 attribute value updates have no impact because the extracted value and the updated value are incorrect. Across the zero-shot, few-shot in-context learning and fine-tuning scenarios, corruptions happen more often than improvements explaining the marginal decrease in performance when self-correction is applied. For instance, in the zero-shot scenario on OA-Mine 19% of the corrections are improvements while 16% of the corrections are corruptions. On both datasets, the amount of corrected values decreases from zero-shot to fine-tuning. For example on OA-Mine, 165 values are updated in the zero-shot scenario while 106 values are updated after fine-tuning. The LLM seems to correct attribute values where it is undecided between two possible values. These undecided attribute values change with in-context learning demonstrations. The updated extracted attribute values of the zero-shot and the few-shot in-context learning scenario overlap only by 13 and 25 values for OA-Mine and AE-110k, respectively.

Tab. 7: Possible outcomes of self-correction.

Outcome	**Attribute**	**Product Description**	**Extracted Value**	**Corrected Value**
Improvement	Net Content	Nature's Path Organic Oatmeal, (Pack of 6, 11.3 Oz Boxes)	11.3 Oz Boxes ×	11.3 Oz ✓
Corruption	Pack size	Goddess Garden - SPF 50 Sunscreen Stick - 1 Unit	1 Unit ✓	n/a ×
Still wrong	Item form	Good Natured Lavender Laundry Soda/Detergent 52 load bag 32 oz.	Soda/Detergent ×	Laundry Soda ×

8 Conclusion

This paper evaluated self-refinement strategies for large language models (LLMs) in the context of product attribute value extraction. The experimental evaluation examined two self-refinement techniques: error-based prompt rewriting and self-correction, across zero-shot, few-shot in-context learning, and fine-tuning scenarios using GPT-4o. Error-based prompt rewriting and self-correction increased computational cost due to higher token consumption without significant gains in extraction performance. While error-based prompt rewriting improved the level of detail of attribute definitions by adding and excluding example values, this likely led to overfitting to the development set. Self-correction occasionally corrected wrongly extracted attribute values but also introduced new errors, leading to insignificant changes in the extraction performance. Overall, fine-tuning without self-refinement achieved the highest F1-score and is the most cost-efficient approach for scenarios where attribute values need to be extracted from a large number of product descriptions.

Bibliography

[BBB24] Brinkmann, Alexander; Baumann, Nick; Bizer, Christian: Using LLMs for the Extraction and Normalization of Product Attribute Values. In: Advances in Databases and Information Systems. pp. 217–230, 2024.

[Br20] Brown, Tom; Mann, Benjamin; Ryder, Nick et al.: Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems. volume 33, pp. 1877–1901, 2020.

[BSB25] Brinkmann, Alexander; Shraga, Roee; Bizer, Christian: ExtractGPT: Exploring the Potential of Large Language Models for Product Attribute Value Extraction. In: Information Integration and Web Intelligence. pp. 38–52, 2025.

[Fa24] Fang, Chenhao; Li, Xiaohan; Fan, Zezhong et al.: LLM-Ensemble: Optimal Large Language Model Ensemble Method for E-commerce Product Attribute Value Extraction. In: Proceedings of the 47th ACM SIGIR International Conference on Research and Development in Information Retrieval. pp. 2910–2914, 2024.

[Go23] Goel, Akshay; Gueta, Almog; Gilon, Omry et al.: LLMs Accelerate Annotation for Medical Information Extraction. In: Proceedings of the 3rd Machine Learning for Health Symposium. pp. 82–100, 2023.

[Hu23] Huang, Jie; Chen, Xinyun; Mishra, Swaroop et al.: Large Language Models Cannot Self-Correct Reasoning Yet. In: Proceedings of the Twelfth International Conference on Learning Representations. 2023.

[Ma23] Madaan, Aman; Tandon, Niket; Gupta, Prakhar; Clark: Self-Refine: Iterative Refinement with Self-Feedback. Advances in Neural Information Processing Systems, 36:46534–46594, 2023.

[Ol23] Olausson, Theo X.; Inala, Jeevana Priya; Wang, Chenglong et al.: Is Self-Repair a Silver Bullet for Code Generation? In: Proceedings of the Twelfth International Conference on Learning Representations. 2023.

[Pa23] Parekh, Tanmay; Hsu, I-Hung; Huang, Kuan-Hao et al.: GENEVA: Benchmarking Generalizability for Event Argument Extraction with Hundreds of Event Types and Argument Roles. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. pp. 3664–3686, 2023.

[Pa24] Pan, Liangming; Saxon, Michael; Xu, Wenda et al.: Automatically Correcting Large Language Models: *Surveying the Landscape of Diverse Automated Correction Strategies*. Transactions of the Association for Computational Linguistics, 12:484–506, 2024.

[Pe23] Peskine, Youri; Korenčić, Damir; Grubisic, Ivan et al.: Definitions Matter: Guiding GPT for Multi-label Classification. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 4054–4063, 2023.

[PR23] Pourreza, Mohammadreza; Rafiei, Davood: DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. Advances in Neural Information Processing Systems, 36:36339–36348, 2023.

[Sh23] Shinn, Noah; Cassano, Federico; Gopinath, Ashwin et al.: Reflexion: language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36:8634–8652, 2023.

[We22] Wei, Jason; Tay, Yi; Bommasani, Rishi et al.: Emergent Abilities of Large Language Models. Transactions on Machine Learning Research, 2022.

[WLJ23] Wang, Xingyao; Li, Sha; Ji, Heng: Code4Struct: Code Generation for Few-Shot Event Structure Prediction. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. volume 1, pp. 3640–3663, 2023.

[Xu19] Xu, Huimin; Wang, Wenting; Mao, Xin et al.: Scaling up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5214–5223, 2019.

[Xu24] Xu, Derong; Chen, Wei; Peng, Wenjun et al.: Large language models for generative information extraction: a survey. Frontiers of Computer Science, 18(6):186357, 2024.

[Ya22] Yang, Li; Wang, Qifan; Yu, Zac et al.: MAVE: A Product Dataset for Multi-source Attribute Value Extraction. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 1256–1265, 2022.

[Zh22] Zhang, Xinyang; Zhang, Chenwei; Li, Xian et al.: OA-Mine: Open-World Attribute Mining for E-Commerce Products with Weak Supervision. In: Proceedings of the ACM Web Conference 2022. pp. 3153–3161, 2022.

[ZYW25] Zhang, Zikang; You, Wangjie; Wu, Tianci: A Survey of Generative Information Extraction. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 4840–4870, 2025.

[ZZH09] Zhang, Liyi; Zhu, Mingzhu; Huang, Wei: A Framework for an Ontology-based E-commerce Product Information Retrieval System. Journal of Computers, 4(6):436–443, 2009.

Workshop on ML4Sys and Sys4ML

Second Workshop on ML for Systems and Systems for ML

Matthias Boehm [1], Andreas Kipf [2], and Manisha Luthra [3]

Abstract: After the first instance at BTW 2023, this second instance of the Workshop on ML for Systems and Systems for ML was held on March 04, 2025 in conjunction with BTW 2025 in Bamberg. The workshop aimed to bring together researchers and practitioners working on applying machine learning (ML) to a spectrum of systems problems as well as building systems for ML pipelines and applications. As a new format, we solicited one-page abstract submissions of already published or ongoing work as well as invited a diverse set of complementary speakers to present prior work. Finally, the workshop featured 12 talks in three sessions and was generally well attended and received.

1 Introduction

The rapid advances in machine learning (ML) have significantly increased its adoption across various fields including data systems, both in academia and industry. These advancements have not only enhanced existing data systems but, in some cases, have completely transformed their internal components, leading to the development of an important field of "learned data system components" in the ML for Systems area. Similarly, a well-structured systems approach has also played a crucial role in advancing current ML techniques and systems, forming the basis for the Systems for the ML area.

Workshop Goals. The workshop aimed to bring together renowned researchers and practitioners through open abstract submissions and invited talks to discuss intriguing topics at the core of the two focus areas. The open format for submitting abstracts from anything like a position paper to technical experiences allowed for lively discussions as well as fostered collaborations among participants. The speakers also shared insights into their ongoing projects and the open research challenges they are currently addressing.

Topics of Interest. In detail, the topics of interest included:

- ML for Systems
  - Learned query processing and optimization
  - Learned index structures and storage layouts
  - Learned algorithms for sorting, compressing, and encoding data
  - Learned data exploration, discovery, and integration
  - Self-tuning and instance-optimized database systems

[1] TU Berlin & BIFOLD, matthias.boehm@tu-berlin.de, https://orcid.org/0000-0003-1344-3663
[2] University of Technology Nuremberg, andreas.kipf@utn.de, https://orcid.org/0000-0003-3463-0564
[3] TU Darmstadt & DFKI, manisha.luthra@dfki.de, https://orcid.org/0000-0002-3788-6664

doi:10.18420/BTW2025-133

  - Learned data systems on emerging hardware and cloud platforms
  - New datasets, benchmarks, and evaluation methods for learned databases
  - Novel use of ML techniques in big data applications
  - Novel use of natural language models and interfaces in data management
  - Building and managing large-scale knowledge bases
- Systems for ML
  - Data and model management in ML applications and complex ML pipelines
  - Data integration, alignment, and preparation of multi-modal training datasets
  - Data cleaning/debugging techniques and data quality management
  - Data augmentation techniques, pipelines, and algorithm integration
  - Data flow optimizations in ML systems (e.g., rewrites, operator fusion)
  - Data- and task-parallel execution strategies for ML pipelines
  - Data access methods in ML systems (e.g., indexing, compression, partitioning)
  - New data system infrastructures and tools for applied ML

2 Organization Committee

Workshop Co-chairs. The workshop was co-organized by the following workshop chairs drawing from their previous experience and papers in this area.

- Manisha Luthra (TU Darmstadt & DFKI)
- Andreas Kipf (University of Technology Nuremberg)
- Matthias Boehm (TU Berlin & BIFOLD)

3 Workshop Format and Program

Workshop Format. Compared to the first instance of the workshop [LKB23], we established a new format of one-page abstract submissions of already published or ongoing work. We received seven of such abstracts, accepted the six of them that matched the focus of the workshop, and additionally invited five complementary talks.

Program. Together, we assembled the following well-balanced program of 12 talks, where submitted abstracts and invited talks got 15+5min and 25+5min speaking time, respectively:

- **Session 1: Systems for ML** (chaired by Matthias)
  - Matthias and Andreas: Opening Remarks
  - Maximilian Schüle (Univeristy of Bamberg): Blue Elephants Inspecting Pandas: Inspection and Execution of Machine Learning Pipelines in SQL
  - Maximilian Böther (ETH Zurich): Modyn: Data-Centric Machine Learning Pipeline Orchestration
  - Stefan Grafberger (TU Berlin): mlwhatif: Data-Centric What-If Analysis for Native Machine Learning Pipelines
- **Session 2: Applications & Benchmarks** (chaired by Matthias)
  - Stefan Hagedorn, Steffen Kläbe (Actian): Experiences of Implementing In-Database TPCx-AI
  - Thaleia-Dimitra Doudali (IMDEA): Keep it Simple, Sustainable! When Is ML Necessary in Cloud Resource Management?
  - Jan-Micha Bodensohn, Liane Vogel (TU Darmstadt & DFKI): Large Language Models for Enterprise Data Engineering
  - Akanksha Vijayvergiya (University Passau): Time-Series Analysis for Life-Science Data
- **Session 3: ML for Systems** (chaired by Andreas)
  - Silvan Reiner (University of Konstanz): ML4DB: Don't Learn What You Already Know
  - Johannes Wehrstein (TU Darmstadt): GRACEFUL: A Learned Cost Estimator For UDFs
  - Giorgio Vinciguerra (University of Pisa): Learned Compression of Nonlinear Time Series With Random Access
  - Immanuel Trummer (Cornell University): CheaPT: Using Language Models Without Breaking the Bank

References

[LKB23] Luthra, M.; Kipf, A.; Boehm, M.: A Tutorial Workshop on ML for Systems and Systems for ML. In: BTW. Pp. 707–708, 2023, DOI: 10.18420/BTW2023-43.

Workshop on Novel Data Management Ideas on Heterogeneous Hardware Architectures

Third Workshop on Novel Data Management Ideas on Heterogeneous Hardware Architectures (NoDMC)

Jan Mühlig[1], Roland Kühn[1], David Broneske[2], and Dirk Habich[3]

The objective of this one-day workshop@BTW 2025 was to explore the challenges and opportunities of data processing on existing and future heterogeneous hardware architectures. Today's processors are no longer mainly bound by the density and frequency of transistors but also by their power and heat budgets. This scenario, often termed the power wall, forces hardware suppliers to design specialized devices optimized for specific computational tasks, resulting in a heterogeneous processor landscape. Consequently, software must explicitly adapt to ensure optimized performance and efficiency across varying hardware configurations.

Additionally, memory and storage has seen an unprecedented change as well: novel and already commercially available techniques have blurred the traditional mental picture of a memory-/storage hierarchy. Technologies like Non-Volatile RAM (NVRAM) challenge the long-standing memory hierarchy prevalent in system-level applications. Moreover, very large caches, High-Bandwidth-Memory (HBM), Non-Uniform Memory Access (NUMA), processing-in-memory (PIM), remote-memory designs, and extremely fast SSDs add to the heterogeneous portfolio of memory/storage techniques. Therefore, to meet the performance requirements of the modern information society, tomorrow's database systems must exploit and embrace this increased heterogeneity of processor and memory technologies.

The third edition of this workshop series served as a forum for discussing ongoing challenges, recent advancements, and future directions in the field. Contrarily to the format of previous editions [BH19; BH23], we put an emphasis on *tutorials* to train and foster a community of researchers and industry practitioners working on data processing challenges on heterogeneous hardware systems. These tutorials offered practical introductions to key concepts, libraries, or frameworks instrumental in engineering and optimizing data-intensive systems on modern heterogeneous hardware, such as (but not limited to) benchmarking, profiling, data parallelism, security considerations, or fault tolerance. Furthermore, the workshop offered a conducive environment for networking, encouraging future collaborations among participants. Especially in view of the fading SPP 2037 on *Scalable Data Management for Future Hardware* and the SPP 2377 on *Disruptive Memory Technologies*, we wanted to strengthen collaborations beyond individual SPP projects by connecting participants of both SPPs with other researchers. To this end, the workshop program was enriched with *lightning talks* and *posters* that explore specialized or emerging hardware technologies, such

1 TU Dortmund University, jan.muehlig@tu-dortmund.de; roland.kuehn@tu-dortmund.de
2 German Centre for Higher Education and Science Studies, DZHW, broneske@dzhw.eu
3 TU Dresden, dirk.habich@tu-dresden.de

doi:10.18420/BTW2025-134

as processing-in-memory devices or racetrack memory, which can be considered as still being nascent. Topics of the workshop included, but were not limited to:

- Applications of modern hardware in data mining, data-intensive machine learning, query processing, sensor or stream processing, or non-traditional applications (e.g., graph processing)
- Algorithms and data structures for efficient data processing on and across different (co-)processors or memory technologies
- Exploitation of specialized ASICs or specialized memories technologies (e.g., PIM)
- Efficient memory management, data placement, and data transfer strategies in heterogeneous systems
- Energy efficiency in heterogeneous hardware environments
- Programming models and hardware abstraction mechanisms for writing data-intensive algorithms on heterogeneous hardware
- Query optimization, cost estimation, and operator placement strategies for heterogeneous hardware
- Transaction processing in heterogeneous systems

After reviewing the received submissions, we were able to accept five tutorials, two lightning talks, and three posters resulting in a full one-day workshop consisting of four sessions.

Overall, the program encompassed perfectly the topical scope of the workshop concept. The workshop started with a keynote by **Philippe Bonnet** from the University of Copenhagen, Denmark on the topic of **Computational Storage (What is it Good For?)**. His keynote opened by defining the principles of computational storage. Then, the goal was to survey the design space and to position the standards in that space. Afterwards, he focused on the lessons learnt from Delilah, a prototypical computational storage platform that implements a variant of the NVMe standard. Finally, he discussed how these lessons apply to current efforts on in- or near-memory processing and co-designed SSDs.

Session1

- **Keynote** by Philippe Bonnet on Computational Storage (What is it Good For?)
- Short **Poster** Presentations
  - Offset-Value Coding using SIMD Intrinsics - Schmeller, Florian (1); Rabl, Tilmann (1); Graefe, Goetz (2) (1: HPI, University of Potsdam; 2: Google, USA)
  - Dynamic Write-Mode Fragmentation for Non-Volatile Memory Simulation - Rau, Janina; Biebert, Daniel; Hakert, Christian; Chen, Jian-Jia (TU Dortmund University)
  - Embracing NVM: Optimizing B^{ϵ}-Tree Structures and Data Compression in Storage Engines - Karim, Sajad (1); Wünsche, Fia (1); Broneske, David (2); Kuhn, Michael (1); Saake, Gunter (1) (1: Otto-von Guericke University Magdeburg; 2: DZHW)

Session2

- **Tutorial:** Understanding Application Performance on Modern Hardware: Profiling Foundations and Advanced Techniques - Mühlig, Jan; Kühn, Roland; Teubner, Jens (TU Dortmund University)
- **Tutorial:** Unleashing the Intel Data Streaming Accelerator - Berthold, André; Schmidt, Lennart; Lehner, Wolfgang; Schirmeier, Horst (TU Dresden)

Session3

- **Tutorial:** Programming Processing-in-Memory for Data Management - Sattler, Kai-Uwe; Jibril, Muhammad Attahir (TU Ilmenau)
- **Lightning Talk:** Feasibility Analysis of Semi-Permanent Database Offloading to UPMEM Near-Memory Computing Modules - Friesel, Birte Kristina; Lütke Dreimann, Marcel; Spinczyk, Olaf (Universität Osnabrück)
- **Lightning Talk:** Lazy DBMS Storage Design with Computational Storage - Baumstark, Alexander; Sattler, Kai-Uwe (TU Ilmenau)

Session4

- **Tutorial:** SIMD for Everyone- A tutorial to TSL - Pietrzyk, Johannes; Krause, Alexander; Lehner, Wolfgang (TU Dresden)
- **Tutorial:** Dreaming of Syscall-less I/O with io uring - Some Assembly Re-quired, Feaver Dreams and Nightmares included - Pestka, Constantin (1); Paradies, Marcus (2) (1: DLR - German Center for Aerospace,Germany; 2: LMU - Munich)

Last but not least, we would like to thank everyone who contributed to this workshop, in particular, the authors, the presenters, the reviewers, the BTW team, and all participants. This workshop was co-organized by the GI-Arbeitskreis *Data Management on Modern Hardware.*

PC Chairs

- Jan Mühlig (TU Dortmund University)
- Roland Kühn (TU Dortmund University)
- David Broneske (German Centre for Higher Education and Science Studies, DZHW)
- Dirk Habich (TU Dresden)

Program Committee

- David Broneske (German Centre of Higher Education and Science Studies, DZHW)

- Patrick Damme (TU Berlin)
- Philipp Götze (SAP SE)
- Juliana Hildebrandt (TU Dresden)
- Roland Kühn (TU Dortmund University)
- Jan Mühlig (TU Dortmund University)
- Hannes Rauhe (Potsdam Institute for Climate Impact Research)
- Annett Ungethüm (TU Hamburg)
- Stefan Wildermann (Friedrich-Alexander Universität Erlangen-Nürnberg)
- Steffen Zeuch (TU Berlin)

References

[BH19] Broneske, D.; Habich, D.: 1st Workshop on Novel Data Management Ideas on Heterogeneous (Co-)Processors (NoDMC). In (Meyer, H.; Ritter, N.; Thor, A.; Nicklas, D.; Heuer, A.; Klettke, M., eds.): Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme"(DBIS), 4.-8. März 2019, Rostock, Germany, Workshopband. Vol. P-290. LNI, Gesellschaft für Informatik, Bonn, pp. 23–25, 2019, DOI: 10.18420/BTW2019-WS-01, URL: https://doi.org/10.18420/btw2019-ws-01.

[BH23] Broneske, D.; Habich, D.: Second Workshop on Novel Data Management Ideas on Heterogeneous (Co-)Processors (NoDMC). In (König-Ries, B.; Scherzinger, S.; Lehner, W.; Vossen, G., 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. Vol. P-331. LNI, Gesellschaft für Informatik e.V., pp. 697–699, 2023, DOI: 10.18420/BTW2023-40, URL: https://doi.org/10.18420/BTW2023-40.

Dynamic Write-Mode Fragmentation for Non-Volatile Memory Simulation

Janina Rau [1], Daniel Biebert [1], Dr.-Ing. Christian Hakert [1], and Prof. Jian-Jia Chen [1]

Abstract: Disruptive memory technology development leads to a wide landscape of novel memory properties. Specifically, with the possibility to utilize multiple write modes for the tradeoff of energy consumption and retention time, the system software and applications can optimize the energy consumption of the memory while ensuring guarantees for memory retention. Considering a system, where individual write modes can be applied to fragments of the total memory space, even larger optimization potentials exist. Deriving online and offline strategies for such optimization requires precise simulation of fragmented memory write modes across the memory area. This paper enables the simulation of such write-mode fragmentation for the NVMain2.0 simulator in a static and dynamic configuration fashion.

Keywords: pcm, gem5, write mode, optimization

1 Introduction

Ongoing exploration of disruptive memory technologies leads to the design proposal of novel methods, which can help to provide more efficient and cost-effective memory in the future. Such methods, however, often provide a trade-off to the system designer or even to the application, when an application-dependent behavior is present. For example, Phase Change Memory (PCM) [Pa14] allows multiple write modes, where individual write modes can be applied in a fragmented manner to certain memory regions. With such a feature, the applications, such as database management systems, can potentially specify how to write a memory portion, resulting in an effect on the retention time, energy consumption, and memory latency. In order to arm software for such memory technologies, strategies and algorithms have to be explored to control fragmented memory write modes in an efficient and effective way.

Related Work Multiple memory write modes, especially for PCM memory, are studied in several approaches for optimization of the memory properties. This ranges from application-specific approaches for neural networks [Si23] utilizing compiler- and runtime-based generic optimization strategies [Li13, Zh19] to generic and formal optimization methods [Pa17].

1 TU Dortmund, Computer Science, Otto-Hahn Straße 16, 44227 Dortmund, Germany,
janina.rau@tu-dortmund.de, https://orcid.org/0009-0003-9848-1999;
daniel.biebert@tu-dortmund.de, https://orcid.org/0009-0006-9061-4189;
christian.hakert@tu-dortmund.de, https://orcid.org/0000-0001-9992-9415;
jian-jia.chen@cs.tu-dortmund.de, https://orcid.org/0000-0001-8114-9760

doi:10.18420/BTW2025-135

While these approaches target different application scopes, they all perform simulations of the effect of different write modes on the energy consumption and retention. The results in [Si23, Zh19] make use of specific extensions to gem5 and NVMain2.0 [PZX15], where direct system extensions, as for instance, specific instructions for different write modes, are realized and the effects are directly simulated. Li et al. [Li13] use a special simulator, which requires specific auxiliary files together with the program to simulate memory effects. Pan et al. [Pa17] utilize specialized versions of NVsim and CACTI in order to simulate the write mode effects.

Our Contributions The previously mentioned studies make use of highly specific in-house simulations to study the effects of their approaches directly. In this paper, in contrast, we aim towards a generic and configurable simulation for fragmented write modes, which enables the experimental evaluation of different optimization approaches.

We show our progress towards equipping the widely-adopted NVMain2.0 simulator [PZX15] with support for energy and latency simulation for statically and dynamically write mode fragmented memory. The source is publicly available on GitHub[2]. We enable the user to configure the memory landscape with different write modes in a user-friendly configuration file. In addition, we provide functional simulation of a CPU interface, which reconfigures the memory write modes during runtime. We conduct experimental evaluation, which highlights the applicability and the possible effect of the write mode simulation extension to NVMain2.0.

2 Write Mode Simulation

Non-volatile memory uses physical properties of the memory materials in order to encode stored memory values. These properties can usually be within a continuous state, where a threshold value defines the logical interpretation of the stored memory value. For instance, to store information, Phase Change Memory (PCM) uses the different resistance states of chalcogenide alloy, i.e., the low resistance crystalline state and the high resistance amorphous state [Bu10]. The percentage of the chalcogenide alloy that is turned to another state depends on the intensity and duration of the electrical programming pulse. Additionally, the cell loses its programmed value over time due to a resistance drift. Hence, the retention time is dependent on the used programming pulse. This leads to the realization of write modes for PCM [Pa14], which provide either a short or long programming pulse. Different write modes therefore result in different energy, retention and latency characteristics of a memory operation. Applying different write modes across the available memory, i.e., realizing a fragmented memory configuration, can minimize the energy consumption and memory latency, while keeping the retention time on the required level. In order to develop schemes for the application of different write modes, simulation tools are crucially required.

2 https://github.com/janinarau/bachelorthesis

In this paper, we propose our realization towards an integration of multiple write modes into the NVM simulator NVMain2.0 [PZX15]. Every memory operation in NVMain2.0 is simulated through a *MemoryRequest* [PZX15], containing all relevant attributes, including the operation type and the memory address. To realize different modes for the energy and latency simulation in a fragmented memory, we equip each request further with an energy and latency attribute, representing the energy consumption and latency of this request. The simulator is further enhanced with a map of memory locations to their corresponding write modes, which defines a pair of energy and latency characteristics for this location. The write mode map is used in order to derive the absolute values of the energy consumption and the latency for each memory request. We realize two approaches to define and modify the write mode mapping in the simulation: A static separation, which is configured upfront to the simulation and remains unchanged and a dynamic separation, which can be modified by the running application during the simulation.

Static Memory Separation To assign different memory write modes to different memory segments, the memory is separated using a configuration file. The user defines each fragment (with any specified size) by specifying the physical location, as well as the corresponding energy and latency values, that are assigned for a memory access in this fragment. At the beginning of the simulation, the configuration file is used to statically define the memory landscape, by transferring the configuration file contents into controller objects, storing and managing the fragments. During the simulation, the controllers map each request by its physical location of memory access to a specific fragment, where the write mode values are used to compute the energy consumption and latency.

Dynamic Memory Separation Allowing the reconfiguration of write modes during runtime enables the development of software-guided strategies for write mode selection. Fixed write modes are defined at the start of the simulation in the form of energy and latency pairs and can be configured during the simulation from the executing application. A CPU interface is realized to carry the information of the desired write mode and corresponding fragment, which is defined by start and end address in the static configuration, to the responsible controller, which issues the reconfiguration.

3 Evaluation

To evaluate the applicability of the fragmented memory simulation, the benchmark *bitcount* from the mibench suite [Gu01] is used, which counts the bits with a value of 1 in an array. For illustrative purposes, we chose a rather simple example to clearly show the effects of static and dynamic memory separation. Fig. 1 depicts the simulation results for the energy consumption of the chosen benchmark. The x axis indicates elapsed simulation time in cycles, the y axis the accumulated energy consumption of the memory device. The blue

dashed line indicates a baseline simulation, where uniform energy and latency characteristics are simulated for the entire memory space. Each SET Operation (0 to 1) is configured to consume $14.03pJ$, every RESET operation (1 to 0) is configured to consume $19.73pJ$, based on the values used in [Ch12]. The green solid line indicates a purely static configured fragmented memory device, where the memory fragments, allocated by the benchmark application, are configured to consume 70% energy and latency of the original write mode. The black dotted line depicts a simulation, where the target application reconfigures the write modes of all memory fragments to 60% at a certain point of the simulation.

It can be observed that the static separation results in a partially different slope of the total energy consumption. Apart from that, the energy consumption behaves similar to the uniform configuration. This highlights that the static memory separation can achieve a detailed simulation of the energy characteristic over the simulation time. The results from the dynamic configuration run show exactly the same behavior for the total energy consumption as the static configuration until the reconfiguration event. Until this point, both simulations apply exactly the same configuration of memory fragments. After the reconfiguration event, it can be observed that the black dotted line continues with a different slope. The overall energy consumption could therefore be lowered, highlighting that the effect of online configurations of the memory can be simulated directly. At the same time a lower energy consumption implies a shorter retention time for the reconfigured memory fragment, which must be taken into account when configuring the write modes.

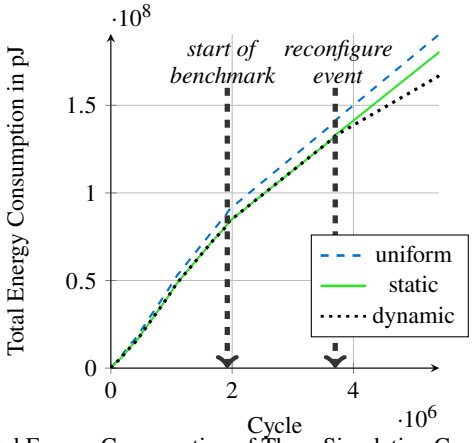


Fig. 1: Total Energy Consumption of Three Simulation Configurations

4 Outlook

Our results are first steps towards a configurable fragmented NVM simulator for different write modes. For future study and completeness of the simulation framework, we will enrich the simulator internals with value-based simulation of the retention of memory objects. Furthermore, we will extend the dynamic interface by further means to control the memory write modes, such as specialized instruction.

Bibliography

[Bu10] Burr, Geoffrey W; Breitwisch, Matthew J; Franceschini, Michele; Garetto, Davide; Gopalakrishnan, Kailash; Jackson, Bryan; Kurdi, Bülent; Lam, Chung; Lastras, Luis A; Padilla, Alvaro et al.: Phase change memory technology. Journal of Vacuum Science & Technology B, 2010.

[Ch12] Chen, Jie; Chiang, Ron C.; Huang, H. Howie; Venkataramani, Guru: Energy-aware writes to non-volatile main memory. SIGOPS Oper. Syst. Rev., 45(3):48–52, January 2012.

[Gu01] Guthaus, Matthew R; Ringenberg, Jeffrey S; Ernst, Dan; Austin, Todd M; Mudge, Trevor; Brown, Richard B: MiBench: A free, commercially representative embedded benchmark suite. In: Proceedings of the fourth annual IEEE international workshop on workload characterization. IEEE, 2001.

[Li13] Li, Qingan; Jiang, Lei; Zhang, Youtao; He, Yanxiang; Xue, Chun Jason: Compiler directed write-mode selection for high performance low power volatile PCM. In: Proceedings of the 14th ACM SIGPLAN/SIGBED conference on Languages, compilers and tools for embedded systems. 2013.

[Pa14] Pan, Chen; Xie, Mimi; Hu, Jingtong; Chen, Yiran; Yang, Chengmo: 3M-PCM: Exploiting multiple write modes MLC phase change main memory in embedded systems. In: Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis. 2014.

[Pa17] Pan, Chen; Xie, Mimi; Yang, Chengmo; Chen, Yiran; Hu, Jingtong: Exploiting multiple write modes of nonvolatile main memory in embedded systems. ACM Transactions on Embedded Computing Systems, 2017.

[PZX15] Poremba, Matthew; Zhang, Tao; Xie, Yuan: Nvmain 2.0: A user-friendly memory simulator to model (non-) volatile memory systems. IEEE Computer Architecture Letters, 2015.

[Si23] Siddhu, Lokesh; Nassar, Hassan; Bauer, Lars; Hakert, Christian; Hölscher, Nils; Chen, Jian-Jia; Henkel, Joerg: Swift-CNN: Leveraging PCM Memory's Fast Write Mode to Accelerate CNNs. IEEE Embedded Systems Letters, 2023.

[Zh19] Zhang, Mingzhe; Zhang, Lunkai; Jiang, Lei; Chong, Frederic T; Liu, Zhiyong: Quick-and-dirty: An architecture for high-performance temporary short writes in MLC PCM. IEEE Transactions on Computers, 2019.

Offset-Value Coding using SIMD Intrinsics

Hardware-accelerated sorting and merging on any modern CPU

Florian Schmeller [1], Tilmann Rabl [1], and Goetz Graefe [2]

Abstract: Core operations in database systems are based on sorting, e.g., creating a new B-tree index, merge joins, or grouped aggregations. The required comparisons can be costly due to many or large columns. In order to reuse previous comparison effort, it can be encoded in form of offset-value codes. While hash values can guarantee that two keys are not equal, offset-value codes can also guarantee equality of keys and indicate their sort order, making them usable in sorting algorithms.

Modern CPUs provide specialized functional units that enable data parallel execution within a single core. In this paper, we report on our initial experiences and measurements for comparisons using SIMD intrinsics. Our techniques are portable to many architectures based on architecture-agnostic vector types and instructions. Our results demonstrate that hardware-accelerated sorting and merging are available on any CPU with SIMD intrinsics, i.e., practically any modern CPU.

Keywords: Database, sorting, merging, comparisons, offset-value coding, longest common prefix, prefix truncation, compression, hardware, acceleration, vector, SIMD

1 Introduction

Sorting is fundamental for constructing B-tree indexes in databases [Co79] and for queries using 'distinct' [BD83], 'group by' [Ep79], and 'join' [BE77]. Interesting orderings are important for effective query optimization [Se79]. Given the relevance of sorting, even the earliest database benchmarks included a sorting component [Bi85]. In the search for performance and efficiency, database research also has a long history in hardware acceleration of database operations [Ai17], including many recent efforts [BZN05, MTA09, Bi16, Lu20]. Recent work on sorting database records utilizes an integer representation of the sort key to leverage SIMD efficiently [Ch08, IT15]. Considering long-deployed and proven sorting hardware [IB88, Iy05] and recent advances in sort-based query processing [GD23], we focus here on efficiently sorting and merging lists of records by caching comparison effort in offset-value codes (OVC) [Co77, DG23], which generalize sorting with longest common prefixes [NK08, BSS20].

Sorting a list of strings involves byte-wise comparisons until a lexicographical order has been established. Replacement selection generates sorted runs from unsorted input, which are

[1] Hasso-Plattner-Institut, Universität Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Deutschland, florian.schmeller@hpi.de, https://orcid.org/0009-0004-7131-926X; tilmann.rabl@hpi.de, https://orcid.org/0009-0009-3335-8045
[2] Google, Madison, Wis., USA, goetzg@google.com, https://orcid.org/0000-0003-0194-6466

doi:10.18420/BTW2025-136

often larger than the allocated memory for the sort operation [Kn73]. As a byproduct during run generation, later merge steps benefit from offset-value coding by reusing these codes to decide comparisons. In fact, IBM mainframes provide the two instructions `UPT` ("update tree") and `CFC` ("compare and form codeword") to implement sorting with tournament trees and offset-value coding [IB88]. As these instructions run only on mainframe CPUs, our goal is hardware acceleration for sorting and merging on any modern CPU with SIMD intrinsics. As a first step towards this goal, this paper focuses on comparisons of normalized keys, i.e., order-preserving binary strings, and on caching comparison effort in offset-value codes.

2 Internal and External Merge Sort using Offset-Value Coding

A simple yet effective compression technique for sorted runs truncates from each database row (or string) the prefix shared with the immediately preceding row. Offset-value coding [Co77] forms an order-preserving surrogate for each row using the prefix size as offset and the data value at that offset as value. The first row in a sorted run has no preceding row with which to share a prefix and its offset is 0.

Merging such runs exploits and refines offsets and compression. In a tree-of-losers tournament tree [Go63] with offset-value coding [Co77], each sub-tree of height n merges 2^n runs using $2^n - 1$ nodes in its binary tree; thus, one run and its key are moved to the root and out of the sub-tree. The key in any sub-tree's root is encoded relative to the escaped key.

The overall winner escapes from the top-most binary node. It has been the winner in each node along its entire path from a leaf node to the root; thus, the entire path is encoded relative to the overall winner. When the winner is written to the merge output and its successor from the same merge input begins its leaf-to-root pass, all keys (the successor and the tree nodes) are coded relative to the prior overall winner.

3 Offset-Value Coding using SIMD Intrinsics

Offset-value code determination. To determine an offset-value code $\mathrm{OVC}(B, A)$ of two binary strings A and B, we first need to the determine the position p, $0 \leq p < \min\{|A|, |B|\}$, of the first difference of A and B, i.e., the length of the longest common prefix $p = |\mathrm{lcp}(A, B)|$. In practice, we compare the two binary strings byte-wise until we find the first two unequal pair of bytes. If we have found p (i.e., the offset), we fetch $B[p]$ as the value. OVCs are a useful byproduct in comparison-based sorting because they represent already computed comparisons and, as a result, reduce memory accesses to key data.

Block-wise vector comparison. We use SIMD intrinsics to speed up the comparison of A and B by loading multiple positions into vector registers and performing block-wise vector comparisons. Fig. 1 illustrates our approach to finding the first difference of A and B using block-wise vector comparisons. The result vector C stores `0xFF` at position i if the two bytes

$A[i]$ and $B[i]$ are equal or `0x00` if they are unequal. A bit mask $\texttt{mask}(C)$ is an unsigned integer where each bit corresponds to the most-significant bit for each byte in C. With this, we can quickly determine if all bytes in C are equal to `0xFF` if $m = \texttt{mask}(C) + 1$, masked with number of bytes in C many bits set to one, equals zero. If not, then we count the trailing bits in m equal to zero to finish the comparison and return p. Otherwise, we continue with the next block and add the vector size to the current offset. In case that the number of bytes to compare n is not evenly divisible by the block size B, we perform a vector comparison with the last $n - B$ bytes if we did not find a difference while processing preceding blocks.

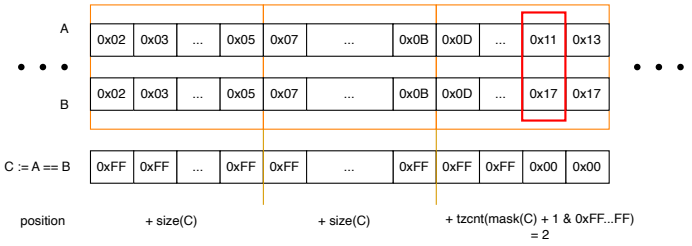


Fig. 1: Block-wise vector comparison.

Algorithm adaptation. Using compiler-supported vector extensions, we can generate platform-agnostic vector code for varying vector sizes with both Clang and GCC [BER23]. Note that while our source code is the same for all platforms (except for concepts that are not supported by Clang and GCC equally), the maximum vector size remains platform-dependent. Therefore, we have to select the working block size based on the requested vectorization support (i.e., the block size supported by the requested vector size) and the number of bytes to compare. We balance the requested block size with the number of bytes to compare such that we can process as many bytes at once as possible. For example, if we have 56 bytes to compare and the platform supports up to 64-byte wide comparisons, we use 32-byte wide comparisons for bytes 0 to 31 and bytes 23 to 55. We use a scalar strategy for comparisons with less than 16 bytes.

Comparing bytes in C. The C standard library defines the `memcmp` function for comparing two byte arrays lexicographically. Determining an offset-value code works similar to `memcmp` except that `memcmp` does not return the position of the first difference. Our vectorization approach is inspired by the vectorized implementation of `memcmp` in the GNU C library (glibc) version 2.36. The maintainers of glibc add vectorization support to `memcmp` for multiple platforms, including x86-64 and AArch64. On x86-64, glibc `memcmp` uses at most 32-byte wide comparisons, employs loop unrolling and optimizes for page crosses[3].

Experimental Evaluation. We conduct our experimental evaluation on a machine with an Intel Cascade Lake CPU (Xeon Gold 5220R), which supports SSE 4.2, AVX2 and AVX-512 instruction set extensions. We compile the benchmark using Clang 18. We load

[3] See https://sourceware.org/git/?p=glibc.git;f=sysdeps/x86_64/multiarch/memcmp-avx2-movbe.S;hb=ae30894. Accessed on February 11, 2025.

two consecutive binary strings from an array of binary strings for their comparison and store the offset in an array of integers. Since the first difference can be at any position, we start the comparison at the beginning of the two binary strings and have the first difference in the last position for each of our evaluation input sizes. With this experiment, we demonstrate the potential of SIMD intrinsics when comparing at least as many bytes as selected in our evaluation, in contrast to a scalar implementation. We align the data to the size of a cache line (i.e., 64 bytes) for input sizes of 64, 128 and 512 bytes and use an alignment of 16 bytes for an input size of 16 bytes.

Fig. 2 shows the results of our benchmark. We report mean values for CPU time in nanoseconds, using log scale. The standard deviation in all micro benchmarks is less than 1% over ten repetitions. As our baseline, *auto* refers to a compiler-optimized scalar loop strategy, which does not generate SIMD intrinsics. We use *vec-128*, *vec-256*, and *vec-512* to denote our vectorization approach with block sizes of 16, 32, and 64 bytes, respectively. On Cascade Lake, Clang uses 256-bit registers when asked to generate 512-bit register code by default[4]. Therefore, we compile with `-mprefer-vector-width=512` to generate code using 512-bit registers. Note that we omit measurements for *vec-256* and *vec-512* on 16 bytes because the requested block size does not match with the input size and will be handled by *vec-128*. Our results show that utilizing SIMD intrinsics improves the comparison performance up to 7.6× for *vec-512* compared to a scalar implementation. We report the measured CPU time for `memcmp` as a red-dotted line. We observe that the performance of our vectorization is on-par with `memcmp`, while additionally outputting offsets for offset-value codes. Our analysis of the workload using `perf` shows that the comparison is I/O bound.

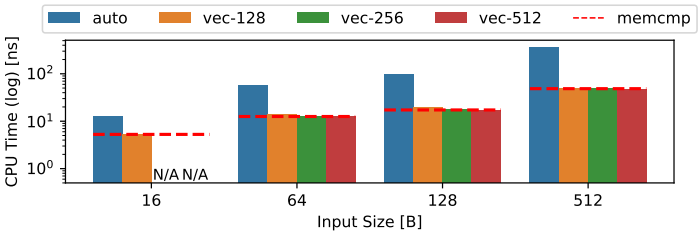


Fig. 2: Comparing two binary strings and return the first difference.

4 Summary and Conclusion

Sorting is ubiquitous in data processing including database query plans and in storage structures including indexes, columnar storage, and their compression. Hardware acceleration for sorting and merging promises broad impact on analytics, stream processing, and data wrangling in preparation of machine learning. Using portable SIMD intrinsics for comparisons, sorting, and merging brings hardware acceleration to many large, important, and expensive workloads. **Acknowledgments.** This work was partially funded by SAP.

[4] See https://godbolt.org/z/W9jx18fsE. Accessed on February 11, 2025.

Bibliography

[Ai17] Ailamaki, Anastasia; Liarou, Erietta; Tözün, Pinar; Porobic, Danica; Psaroudakis, Iraklis: Databases on Modern Hardware: How to Stop Underutilization and Love Multicores. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2017.

[BD83] Bitton, Dina; DeWitt, David J.: Duplicate Record Elimination in Large Data Files. ACM Trans. Database Syst., 8(2):255–265, 1983.

[BE77] Blasgen, Mike W.; Eswaran, Kapali P.: Storage and Access in Relational Data Bases. IBM Syst. J., 16(4):362–377, 1977.

[BER23] Benson, Lawrence; Ebeling, Richard; Rabl, Tilmann: Evaluating SIMD Compiler-Intrinsics for Database Systems. 14th International Workshop on Accelerating Analytics and Data Management Systems Using Modern Processors and Storage Architectures, 2023.

[Bi85] Bitton, Dina; Brown, Mark P.; Catell, Rick; Ceri, Stefano; Chou, Timothy C. K.; DeWitt, David J.; Gawlick, Dieter; Garcia-Molina, Hector; Good, Bob; Gray, Jim; Homan, Pete; Jolls, Bob; Lukes, Tony; Lazowska, Ed; Nauman, John; Pong, Mike; Spector, Alfred Z.; Trieber, Kent; Sammer, Harald; Serlin, Omri; Stonebraker, Michael; Reuter, Andreas; Weinberger, Peter J.: A measure of transaction processing power. Datamation archive, 31:112–118, 1985.

[Bi16] Binnig, Carsten; Crotty, Andrew; Galakatos, Alex; Kraska, Tim; Zamanian, Erfan: The End of Slow Networks: It's Time for a Redesign. Proc. VLDB Endow., 9(7):528–539, 2016.

[BSS20] Bingmann, Timo; Sanders, Peter; Schimek, Matthias: Communication-Efficient String Sorting. In: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, May 18-22, 2020. IEEE, pp. 137–147, 2020.

[BZN05] Boncz, Peter A.; Zukowski, Marcin; Nes, Niels: MonetDB/X100: Hyper-Pipelining Query Execution. In: Second Biennial Conference on Innovative Data Systems Research, CIDR 2005, Asilomar, CA, USA, January 4-7, 2005, Online Proceedings. www.cidrdb.org, pp. 225–237, 2005.

[Ch08] Chhugani, Jatin; Nguyen, Anthony D.; Lee, Victor W.; Macy, William; Hagog, Mostafa; Chen, Yen-Kuang; Baransi, Akram; Kumar, Sanjeev; Dubey, Pradeep: Efficient implementation of sorting on multi-core SIMD CPU architecture. Proc. VLDB Endow., 1(2):1313–1324, 2008.

[Co77] Conner, W. M.: Offset-value coding. IBM Techn. Disclosure Bull., p. 2832–2837, 1977.

[Co79] Comer, Douglas: The Ubiquitous B-Tree. ACM Comput. Surv., 11(2):121–137, 1979.

[DG23] Do, Thanh; Graefe, Goetz: Robust and Efficient Sorting with Offset-value Coding. ACM Trans. Database Syst., 48(1):2:1–2:23, 2023.

[Ep79] Epstein, Robert: Techniques for Processing of Aggregates in Relational Database Systems. Technical Report UCB/ERL M79/8, University of California, Berkeley, Feb 1979.

[GD23] Graefe, Goetz; Do, Thanh: Offset-value coding in database query processing. In: Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023. OpenProceedings.org, pp. 464–470, 2023.

[Go63] Goetz, Martin A.: Internal and tape sorting using the replacement-selection technique. Commun. ACM, 6(5):201–206, 1963.

[IB88] IBM: Enterprise System Architecture/370, Principles of Operation. IBM publication SA22-7200-0, 1988.

[IT15] Inoue, Hiroshi; Taura, Kenjiro: SIMD- and Cache-Friendly Algorithm for Sorting an Array of Structures. Proc. VLDB Endow., 8(11):1274–1285, 2015.

[Iy05] Iyer, Balakrishna R.: Hardware Assisted Sorting in IBM's DB2 DBMS. In: Advances in Data Management 2005, Proceedings of the 12th International Conference on Management of Data, COMAD 2005b, December 20-22, 2005, Hyderabad, India. Computer Society of India, 2005.

[Kn73] Knuth, Donald E.: The Art of Computer Programming, Volume III: Sorting and Searching. Addison-Wesley, 1973.

[Lu20] Lutz, Clemens; Breß, Sebastian; Zeuch, Steffen; Rabl, Tilmann; Markl, Volker: Pump Up the Volume: Processing Large Data on GPUs with Fast Interconnects. In (Maier, David; Pottinger, Rachel; Doan, AnHai; Tan, Wang-Chiew; Alawini, Abdussalam; Ngo, Hung Q., eds): Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020. ACM, pp. 1633–1649, 2020.

[MTA09] Müller, René; Teubner, Jens; Alonso, Gustavo: Data Processing on FPGAs. Proc. VLDB Endow., 2(1):910–921, 2009.

[NK08] Ng, Waihong; Kakehi, Katsuhiko: Merging string sequences by longest common prefix. In: IPSJ Digital Courier. volume 4, pp. 69–78, 2008.

[Se79] Selinger, Patricia G.; Astrahan, Morton M.; Chamberlin, Donald D.; Lorie, Raymond A.; Price, Thomas G.: Access Path Selection in a Relational Database Management System. In: Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data, Boston, Massachusetts, USA, May 30 - June 1. ACM, pp. 23–34, 1979.

Embracing NVM: Optimizing B$^\epsilon$-Tree Structures and Data Compression in Storage Engines

Sajad Karim [1], Fia Wünsche [1], David Broneske [2], Michael Kuhn [1], and Gunter Saake [1]

Abstract: Non-volatile memory (NVM) introduces a new class within the traditional storage hierarchy, combining attributes of both primary and secondary storage. NVM technologies offer latency approaching that of DRAM, though slightly higher[3], and significantly lower than traditional block storage devices. NVM is byte-addressable and provides persistence. Integrating NVM into various database and file systems has been the focus of extensive research. Efforts have centered on optimizing data structures like B-trees and LSM-trees and enhancing components such as buffer manager, logging, and recovery. However, a unified storage engine specifically designed to study the characteristics of the modern storage landscape, effectively utilizing and managing the potential of diverse memory and storage devices, does not currently exist. In this project, SMASH, we extend a storage engine to explore the benefits of the modern storage landscape. By integrating NVM into its storage stack, we aim to optimize data structures and employ compression techniques to minimize memory footprint. Initial experiments show strengths under sequential workloads and sensitivity to block sizes. Further modifications enable zero-copy deserialization and granular key-value access, fully leveraging NVM's byte-addressable characteristics.

Keywords: Non-Volatile Memory, NVM-optimized B$^\epsilon$-tree, Object and Key/Value Storage Engine

1 Introduction

The rapid growth of data-intensive applications coupled with the introduction of heterogeneous memory and storage technologies has exposed critical limitations in existing database management systems (DBMS) and file systems. Current systems rely on volatile caches and traditional block-based storage models that fail to utilize emerging NVM technologies efficiently [Ko23]. This results in suboptimal performance, inconsistent data management, and limited adaptability in the context of modern heterogeneous storage architectures. Furthermore, existing solutions often focus on specific devices, missing the opportunity to leverage a combination of storage technologies to address varying application needs.

To address these challenges, the SMASH project proposes a common storage engine for the modern heterogeneous storage landscape. SMASH uses B$^\epsilon$-trees as an index structure. Unlike other B-tree variants designed to enhance query performance, B$^\epsilon$-trees achieve substantially higher insert throughput, making them ideal for workloads that involve intensive writes. SMASH, with B$^\epsilon$-tree, aims to intelligently manage data placement, retrieval, and migration across diverse storage devices such as NVM, SSDs, and HDDs. Additionally, SMASH introduces native support for data transformations like compression.

[1] Otto von Guericke University, Magdeburg, Germany, sajad.karim@ovgu.de, 0009-0002-4910-8453; fia.wunsche@ovgu.de, 0000-0002-5304-7262; michael.kuhn@ovgu.de, 0000-0001-8167-8574; saake@ovgu.de, 0000-0001-9576-8474
[2] DZHW, Hannover, Germany, broneske@dzhw.eu, 0000-0002-9580-740X
[3] NVM technologies such as MRAM [Sl02], PCM [FNW17], and RRAM [Ch20] each have distinct latencies.

doi:10.18420/BTW2025-137

As a significant contribution, SMASH will deliver a flexible and reusable software library to support database and high-performance computing (HPC) workflows, addressing modern data systems' performance and scalability challenges.

The SMASH project is a collaborative effort involving different teams working on the abovementioned challenges. While the project as a whole focuses on developing a common storage engine, our team's primary research is focused on two key areas: first, data structure and data management, and second, data compression, which leads to the following objectives:

- Data structure and data management: To optimize the B^ϵ-tree data structure for efficient utilization of heterogeneous memory and storage devices (DRAM, NVM, SSDs, and HDDs) by enabling fine-grained data management and leveraging NVM's unique characteristics for improved performance.
- Data compression: To integrate efficient data compression techniques optimized for NVM to reduce memory and storage overhead and improve read efficiency without compromising performance.

These objectives guide our approach to extending the storage engine by modifying the data structure for granular key-value access, minimizing memory footprint, and zero-copy deserialization, leveraging NVM's byte-addressability and low latency. We are also introducing compression strategies tailored for NVM. Benchmarking for transactional and analytical workloads will demonstrate the storage engine's real-world effectiveness, guiding further optimization. These tasks collectively address critical gaps, enhancing data management efficiency, scalability, and adaptability for database applications.

2 The Current State of Our Research

Haura: SMASH utilizes a prototype storage engine called Haura[4] to manage data across different storage hierarchies. Haura operates as a tiered storage stack in user space, using a B^ϵ-tree index for efficient data retrieval on block storage devices. Built with a modular architecture, it supports advanced features such as snapshots, data placement strategies, and fail-over mechanisms, ensuring reliability and high performance.

NVM's Integration into Haura: The initial phase focused on integrating NVM into Haura's storage stack by adapting the architecture. Modifications have been made in each module of Haura to support NVM, thereby ensuring that relevant hints and settings are seamlessly passed to the bottom layer interfacing with NVM [Ka23]. Following the NVM integration, a comprehensive evaluation of Haura was conducted [Ka23]. Micro-benchmarks have been conducted to assess Haura's performance, focusing on the impact of NVM in both sequential and random access patterns. The outcomes reveal that NVM performs optimally under sequential workloads while random access performance is enhanced with larger block sizes. Sequential workloads exploit Optane DCPMM's (i.e., the NVM used in our study) bandwidth, low latency, and prefetching and parallelism capabilities, while larger block sizes mitigate random access inefficiencies by reducing per-operation overhead. Additionally,

[4] https://github.com/parcio/haura

the experiments included testing various configurations, such as different thread counts and cache sizes, to understand the performance further. The results indicate that smaller block sizes favor NVM while larger block sizes benefit NVMe SSD, highlighting the significant influence of block size on overall performance. Furthermore, various libraries for accessing NVM using PMDK have been evaluated, with libpmem selected for its consistent performance across different buffer sizes [Ka23].

Data Access Optimizations for NVM: In the NVM integration work, although different node sizes have been used for NVM and block storage devices, the entire block still had to be copied to DRAM to read its content, thus not optimized for NVM. To fully utilize NVM's characteristics, further modifications have been made to leverage NVM's byte-addressability, enabling direct data to be read from NVM without copying the entire node to DRAM. Additionally, zero-copy deserialization has been incorporated to reduce the memory footprint further. The benchmarking of optimizations is in progress. Moreover, given that Haura is a well-established prototype engine, making changes at a layer is complex and time-consuming. Therefore, a proof-of-concept (POC) project[5] has been created to work on various optimizations. The project acts as a sandbox for granular analysis before moving changes to Haura. The proposed modifications aim to leverage NVM's persistence, byte-addressability, and low latency to improve tree performance and reduce computational overhead, particularly by performing direct reads on NVM and minimizing the need for tree rebalancing operations. Figure 1 depicts the benefit of reading only necessary data instead of whole nodes. In the experiment, data in nodes on NVM, DRAM, and NVMe SSD have been accessed in different settings, including partial reads like `int`s and `substring`s (the sizes are mentioned on the horizontal axis). The cases where only a few bytes are accessed imitate scenarios where only the required data is accessed. Evidently, the cost for nodes on NVMe SSD is high, as the entire node must be fetched into DRAM regardless of the access type and the actual bytes necessary for the operation.

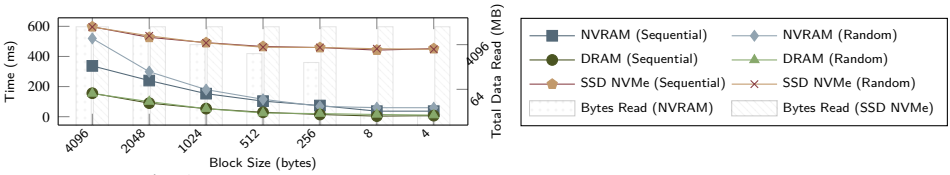


Fig. 1: Impact of byte-granular partial reads on NVM versus full node access.

Compression: Data transformation is another important aspect of the project. The project aims to natively support data transformations, such as compression, enhancing performance and reducing storage overhead. The goal is to optimize data storage across heterogeneous storage hierarchies while maintaining flexibility and high performance. By supporting compression, SMASH improves storage efficiency and data transfer throughput. It integrates lossless compression directly into the system.

Due to the fundamental differences between NVM and traditional block devices, updating the compression logic becomes necessary. Unlike block storage, which operates on larger,

[5] https://github.com/SajadKarim/haldendb

fixed-size blocks, NVM is byte-addressable, allowing for more granular data operations. The conventional approach of compressing entire nodes or blocks impacts the physical and logical offsets of data, necessitating the copying of the block back to DRAM, thereby undermining NVM's byte-addressability capabilities. Consequently, an optimized approach is required to fully leverage NVM. The NVM-optimized tree allows for more granular reads, such as key/value access, but compressing the whole node hinders this capability. Thus, achieving compression at the data structure level, specifically at the key/value level, permits the access of only the desired values within a node, significantly enhancing read efficiency and overall data access performance.

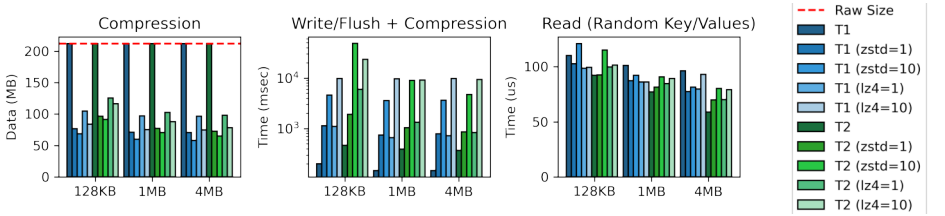


Fig. 2: Performance comparison of block and NVM-optimized B$^\epsilon$-trees using Silesia compression corpus with Zstd and LZ4 algorithms.

In Figure 2, the Silesia compression corpus is utilized for writing to block (T1) and NVM-optimized B$^\epsilon$-tree (T2) through the object API, with varying node and object chunk sizes. The corpus comprises a diverse collection of files representing typical data encountered in real-world scenarios, serving as a standard dataset for assessing the efficacy and efficiency of lossless compression techniques. Zstd and LZ4 compression algorithms are employed; Zstd provides high compression ratios, while LZ4 prioritizes speed. The object API splits objects (files) into chunks, impacting compression ratios based on node/chunk sizes. Expectedly, higher compression levels (e.g., Zstd=10) require significantly more time than lower levels (e.g., Zstd=1, LZ4=1). Performance metrics show that the NVM-optimized B$^\epsilon$-tree (T2) demonstrates superior read numbers compared to the block tree (T1), with node/chunk sizes affecting read efficiency.

3 Conclusion

In conclusion, this ongoing research highlights the potential of NVM in optimizing modern storage systems. Integrating NVM into the Haura engine has already demonstrated notable performance benefits, particularly under sequential workloads and when leveraging NVM's byte-addressability. Tailoring compression logic to NVM's granularity has further enhanced read efficiency by enabling key-value level access. However, the work remains in progress, with future efforts focused on optimizing the B$^\epsilon$-tree and positioning the index structure with its keys compressed on NVM. These efforts will ensure that the proposed optimizations deliver a robust, flexible, and efficient storage engine capable of addressing the growing demands of heterogeneous storage hierarchies in data-intensive applications.

Acknowledgement

This work is part of SPP 2377 and was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 502268500.

Bibliography

[Ch20] Chen, Yangyin: ReRAM: History, status, and future. IEEE Transactions on Electron Devices, 67(4):1420–1433, 2020.

[FNW17] Fong, Scott W; Neumann, Christopher M; Wong, H-S Philip: Phase-change memory - Towards a storage-class memory. IEEE Transactions on Electron Devices, 64(11):4374–4385, 2017.

[Ka23] Karim, Sajad; Wünsche, Johannes; Broneske, David; Kuhn, Michael; Saake, Gunter: Assessing non-volatile memory in modern heterogeneous storage landscape using a write-optimized storage stack. Grundlagen von Datenbanken, 2023.

[Ko23] Koutsoukos, Dimitrios; Bhartia, Raghav; Friedman, Michal; Klimovic, Ana; Alonso, Gustavo: NVM: Is it Not Very Meaningful for Databases? Proceedings of the VLDB Endowment, 16(10):2444–2457, 2023.

[Sl02] Slaughter, JM et al.: Fundamentals of MRAM technology. Journal of superconductivity, 15(1):19–25, 2002.

Understanding Application Performance on Modern Hardware: Profiling Foundations and Advanced Techniques

Jan Mühlig [1], Roland Kühn [1], and Jens Teubner [1]

Abstract: Engineering performance-sensitive applications necessitates a deep comprehension of the interactions between hardware and software. Although profiling tools are available to assist, they often struggle to precisely analyze specific segments of an application. And, crucial information, such as data object addresses, is challenging to relay from the application to external tools. This tutorial demonstrates how these challenges can be addressed by using libraries that enable performance profiling directly within the application.

1 Introduction

Understanding the interplay between software and hardware is paramount for optimizing the performance of data-intensive applications, such as database management systems (DBMSs). However, the architecture of modern hardware has grown increasingly complex—including vast cache hierarchies, sophisticated memory systems, and advanced CPU cores equipped with features like out-of-order execution, intricate branch prediction algorithms, and simultaneous multi threading. This complexity introduces substantial challenges for engineers trying to decipher the interaction of hardware and software, making *performance profiling* a critical practice to pinpoint and—ideally—mitigate performance bottlenecks.

A variety of tools, such as Intel *VTune* [In24b], AMD *μProf* [AM24], and Linux *Perf* [DM10], support engineers in analyzing execution performance. Under the hood, these tools utilize *performance counters*: dedicated registers baked into nearly all modern processors that monitor low-level hardware events, such as the count of executed instructions or cache hits and misses. By correlating these events with instruction pointers, the tools can help to diagnose resource-intensive lines of code.

However, Linux Perf and suchlike—being external—typically profile the entire application, complicating the targeted profiling of specific application segments. This limitation is particularly problematic when analyzing micro-benchmarks, where the benchmark itself may represent only a fraction of the overall runtime, or when attempting to distinguish between different operational phases, such as the fill phase versus the lookup phase in, for instance, tree benchmarks. Additionally, while these tools can correlate events with higher-level constructs like code lines using the binary's symbol names, they often miss the

[1] DBIS Group, TU Dortmund University, Dortmund, Germany,
jan.muehlig@tu-dortmund.de, https://orcid.org/0009-0008-2226-6367;
roland.kuehn@cs.tu-dortmund.de, https://orcid.org/0000-0002-1485-140X;
jens.teubner@cs.tu-dortmund.de, https://orcid.org/0000-0002-0344-5203

doi:10.18420/BTW2025-138

opportunity to leverage deeper, application-specific insights, such as the memory location of data structure instances [No20]. Consequently, while it is possible to identify instructions exhibiting high memory access latency, linking them to specific instances of data structures remains elusive. Consider tree-like data structures as an example: Different tree levels experience unique access patterns; nodes logically close to the root tend to stay in the CPU cache, whereas nodes close to the leaves, accessed less frequently, are more likely to be evicted. Since nodes are typically accessed using the same code, it becomes challenging to pinpoint the *true* source of high access latency, which is primarily the node instance rather than the line of code.

In this tutorial, we demonstrate how to address these challenges by integrating profiling directly into (C++) applications. This integration allows for fine-grained profiling and the injection of application-specific insights into the profiling results, thereby enhancing our understanding of the interactions between software and hardware. The tutorial is structured as follows: Section 2 provides background on performance profiling techniques. Section 3 illustrates how applications can manage hardware events, using a B^+-tree as a running example. Finally, Section 4 outlines the timeline and structure of this tutorial.

2 Using Performance Counters to Understand Hardware/Software Interaction

Modern processors provide two different methods to gain an understanding of how software interacts with the underlying hardware substrate. First, Performance Monitoring Units (PMUs) can count the occurrence of low-level hardware events like cache misses throughout the software's execution cycle. Second, processors can periodically capture snapshots that offer more fine-grained data, such as the currently executing instruction and the accessed memory address.

2.1 Counting Hardware Events

For counting event statistics, PMUs are deployed in a timer-like fashion: They are programmed with specific events to count, started before the execution of code to monitor, and stopped subsequently. The difference in recorded values before and after the execution reflects the events' occurrence during the software's operational phase. This practice of recording key performance indicators is both well-known and extensively used in professional system engineering and academic research (e.g., [LLS13, ZF15, Be21, Sc23, Kü23, KMT24]).

Modern hardware offers numerous PMUs that can record events simultaneously, with the range of monitorable events expanding significantly in recent processor generations. However, this event diversity varies by CPU generation and manufacturer, complicating performance comparisons across chipmakers.

2.2 Detailed Analysis through Sampling

Although coarse-grained statistics can highlight inefficient executions, they seldom pinpoint the exact origin of bottlenecks, such as which specific instruction causes cache misses or branch mispredictions. For a more granular analysis, modern processors are equipped with *sampling*-based mechanisms that periodically capture the current state of execution. The general idea is to specify one or multiple *trigger* events along with a threshold; the PMU will then count the trigger event and fire a sample upon reaching the threshold, for instance, every 4 000th CPU cycle. Users can also dictate the specifics included in the sample, such as the pointer of the currently executing instruction, the accessed memory address, details of the access like latency and data origin, register values, and even branch and call stacks.

The most common method of utilizing these samples is *code-based* profiling, implemented by various profiling tools such as *VTune*, AMD *μProf*, and Linux *Perf* (particularly through the Perf *record* subcommand). These tools focus on monitoring instruction pointers and correlating them with higher-level programming constructs, e.g., `C++` functions and lines of code. This enables the identification of code that is executed frequently or consumes significant execution time. The derived insights can pinpoint specific sections of code that are critical bottlenecks, such as those causing CPU stalls while waiting for data transfers from the memory subsystem.

2.3 Abstractions

Different CPU manufacturers implement sampling mechanisms in distinct ways [Sa23]. For instance, since the Nehalem architecture, Intel implements Processor Event-Based Sampling (PEBS) [In24a, AH17, Yi20], which allows the PMU to leverage almost any configurable event as a trigger for sampling. When the specified threshold is reached, the CPU captures a snapshot and stores the sampled data into a dedicated hardware-managed buffer. As this buffer fills, the CPU interrupts the kernel to transfer the samples into a user-level buffer for further analysis.

AMD's Instruction-Based Sampling (IBS) [Dr07, Dr10] works differently: Each CPU core has two specific PMUs that can be used to sample either *instruction fetching* or *execution* events—the latter based on executed CPU cycles or micro-operations [Sa23]. To that end, the CPU counts either fetched instructions or executed cycles/micro-operations until the threshold is met. Upon reaching this threshold, the CPU traces the next instruction through the entire fetch or execution pipeline, allowing for the collection of diverse data types, such as TLB statistics from instruction fetches or memory-based statistics from execution monitoring. Since the ARMv8.2 ISA extension, ARM implements Statistical Profiling Extension (SPE), a sampling procedure similar to Intel PEBS and AMD IBS [Mi24]. However, SPE is an optional feature, usually only implemented in high-performance cores like the Neoverse N1 or the Cortex-A78.

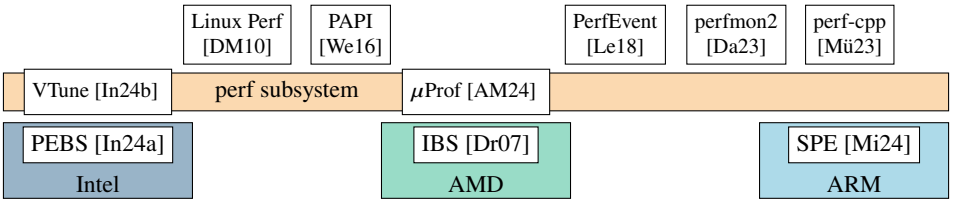


Fig. 1: Abstraction of different profiling approaches. The *perf subsystem* generalizes the utilization of both across different CPUs and manufacturers, enabling several libraries to control PMUs.

These methodological differences extend to the configuration of the hardware and interpretation of results. Hence, manufacturers typically develop tailored profiling tools, such as Intel's VTune and AMD's μProf. To bridge these variations, the Linux kernel offers a unified solution through the *perf subsystem*, providing a standardized interface for recording statistics and sampling across various CPUs and manufacturers. This subsystem is deeply integrated within the kernel architecture and forms the foundation of the Linux *Perf tool*, which is designed to handle both the recording of performance statistics and detailed event sampling. Fig. 1 illustrates the abstraction levels corresponding to various manufacturer-specific examples.

3 Fine-granular Performance Monitoring

The mentioned instruments wrap the counting of events and the recording of samples throughout an application's entire execution. This broad approach limits their effectiveness in profiling, for example, counting cache misses only within a specific code segment.

3.1 Phase-wise Event Counting

While the perf subsystem allows controlling PMUs from the application, its interface is notably complex and challenging to utilize effectively. As a response to these challenges, recent years have seen the development of several abstractions on top of the perf subsystem to record performance counters and samples, e.g., [We16, Da23, Le18, Mü23] (see Fig. 1).

One example of such an abstraction is the *perf-cpp* [Mü23] library, which will be used as an illustrative application in this tutorial. Similar to the Perf *stat* subcommand in Linux Perf, perf-cpp enables to specify hardware events for counting. However, in contrast to Linux Perf, perf-cpp provides the capability to control profiling directly from within the application, which enables monitoring of specific code regions. For instance, when benchmarking a tree-like data structure, it is common practice to exclude data generation from performance measurements and to monitor individual phases, such as the fill and lookup phases, separately.

```
1  tree = Tree()
2  event_counter = EventCounter()
3  event_counter.add("cycles","instructions","cache-misses")
4  insert_workload = generate_insert_workload()      // Generate workload data for inserts
5  lookup_workload = generate_lookup_workload()                          // and lookups
6  foreach tuple in insert_workload
7  |  tree.insert(tuple)                                                 // Populate the tree
8  event_counter.start()                        // Wrap event counting around the lookup phase
9  foreach key in lookup_workload
10 |  tree.lookup(key)
11 event_counter.stop()
12 results = event_counter.result()                  // Consume results for further processing
```

Fig. 2: Profiling only the lookup-phase of a tree benchmark using perf-cpp.

Fig. 2 demonstrates the use of perf-cpp in this context: Following workload generation (lines 4 and 5), the tree is populated with data (lines 6 and 7). Event counting begins just before the lookup phase and ends thereafter (lines 8–11), enabling precise measurement of only this phase. Upon executing all lookups, the results of the monitored hardware events (see line 3) can be consumed and processed for further analysis (line 12).

3.2 Access Analysis via Sampling

While event counting provides an initial start-to-finish insight into the interaction between hardware and software of specific code segments, it lacks details on how this interaction evolves throughout the execution. Take, for instance, the case of a tree-like structure: Trees are typically plagued by *pointer-chasing*, where the next node to visit during a traversal is accessed directly after identification—leaving no space for the hardware- or software-prefetcher to jump in. To address this challenge, the use of *coroutines* enables pipelining of node accesses, effectively introducing a time gap between identifying and accessing a node. This interval allows for transferring nodes closer to the CPU via *software-based prefetching* [Jo18, Ps19, HLW20, MT21].

Challenges. Although pipelining plus prefetching reduces CPU cycles spent waiting for data to arrive, these stalls have not been eliminated entirely in former research. Sampling can offer deeper insights to identify remaining bottlenecks. However, tools like Linux Perf and Intel VTune predominantly rely on code-based profiling: mapping sampled instruction pointers to lines of code. This profiling can pinpoint which code experiences high memory latency, though it provides a somewhat skewed view; access penalties vary across the tree—"higher" benefit from cache locality than deeper leaf nodes. While Linux Perf integrates

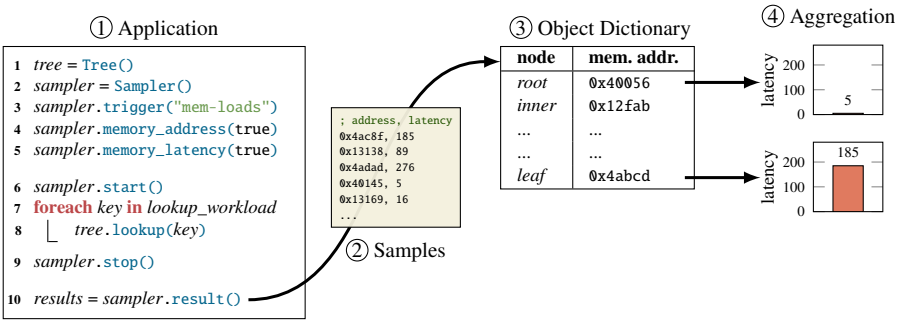


Fig. 3: Sampling tree-lookups within the application and mapping samples to tree nodes.

memory-based sampling, it is limited by its inability to correlate sampled memory addresses with application-specific data structures, such as tree nodes.

Pinpointing Memory Latency. This is where in-application sampling becomes valuable: By capturing access details such as memory addresses and latency directly within the application, this method not only enables precise, fine-grained profiling of specific code segments but also allows to enrich these samples with application-specific knowledge.

Fig. 3 outlines this approach using perf-cpp: First, the lookup phase of a tree benchmark is monitored by initiating the sampler prior to executing lookups and stopping it afterward (①). The gathered data (②) are subsequently correlated with specific tree nodes using a dictionary that catalogs all node instances (③). Once all samples are accurately mapped to their respective nodes, the average access latency for each tree node can be computed (④).

Analyzing Software Prefetching for B$^+$-trees. We will now zoom into a detailed analysis driven by in-application sampling, focusing on software prefetching techniques within B$^+$-trees. To take a close look, we augmented a well-established B$^+$-tree implementation [LHN19] with coroutine-based pipelining and software-based prefetching techniques as proposed by Psaropoulos et al. [Ps19]. For our experiments, we utilized the YCSB benchmark [Co10] with 100 million records and lookup requests, executing the benchmark on a single thread. Note that we used different node sizes: 4 kB nodes yield the best performance for the unmodified B$^+$-tree, while software-prefetching necessitates smaller nodes to avoid overwhelming the LFB [KMT24]. Therefore, for the prefetching version, we employed nodes sized at 256 B.

As outlined in Fig. 3, we configured perf-cpp to sample exclusively during the lookup phase and requested to include memory addresses and latency details in the samples. However, in contrast, each sample was not only mapped to a distinct tree node but to a specific data segment within that node. This allows to aggregate memory latency data for different

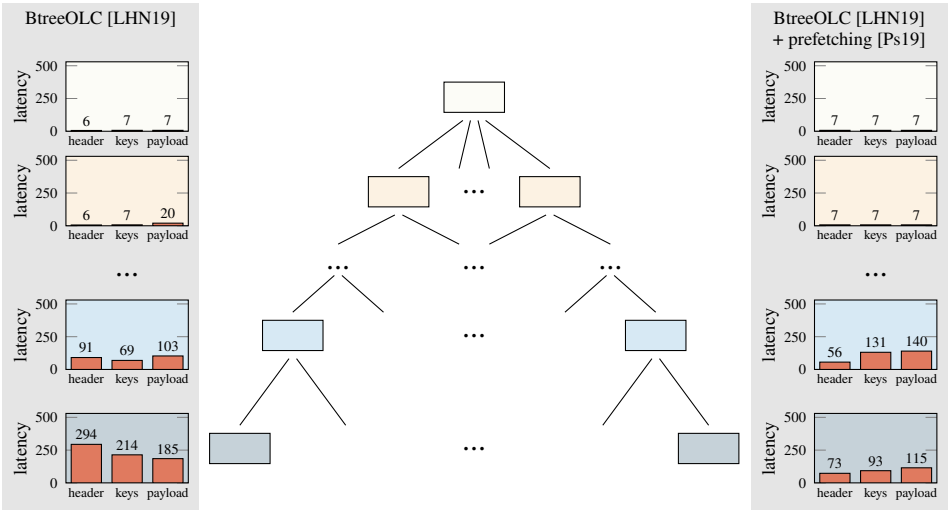


Fig. 4: Memory access latency for different node-segments (header, keys, payloads) in various tree levels derived from sampling—*without* prefetching (left) and *with* software-based prefetching (right).

segments of a node: the *header* (typically accessed first), the *keys* segment, and the *payload* containing the value or child pointer.

Fig. 4 presents the results, detailing the average access latency (in CPU cycles) for each node segment at the top two and bottom two levels of the tree. The left side of the figure illustrates the latency for the original B$^+$-tree implementation *without* prefetching, while the right side shows the outcomes with prefetching applied. Unsurprisingly, in both scenarios, the root node exhibits cache locality, with latency comparable to those of the L1 data cache—the same applies to the root's straight child nodes. In contrast, the deeper levels, such as the leaves, exhibit significantly higher access latency, reaching up to 290 cycles when prefetching is not implemented. With prefetching applied, the latency at the leaf node level is significantly reduced, effectively doubling the throughput. Nonetheless, the experiment indicates that many node accesses still experience noticeable latency.

Consequently, these insights can draw a range of optimizations to improve prefetching in tree-like data structures. First, since the upper levels already benefit from cache locality, prefetching these nodes is redundant. By omitting prefetching for nodes already sitting in the cache, we can reduce instruction bandwidth spent on executing prefetch instructions and minimize the overhead associated with coroutines: Nodes visited consecutively without prefetching involved can be accessed within a single coroutine. In subsequent experiments, we saw a modest improvement of approximately 4% when applying this optimization for the first three tree levels. Note that the effectiveness of this optimization is heavily influenced by the cache and the size of the tree nodes.

Second, the findings suggest that prefetching is not being applied optimally, potentially due to the *prefetch distance*—the time gap between executing the prefetch instruction and the actual data access. In our implementation, this interval is dictated by a round-robin coroutine scheduler (as described in [Ps17]) and a *fixed* number of coroutines[2] active simultaneously: Each coroutine is executed after all others yield the control flow. However, data movement from memory into caches varies in duration; similarly, coroutines at different stages of a traversal have variable execution times (particularly with the mentioned optimization in place). Accordingly, the interval between prefetch and access varies and is hard to control. This challenge becomes further complex when considering operations beyond lookups that might experience more *heterogeneous* execution times. We believe that, to enhance the effectiveness of prefetching combined with coroutines, adopting a more refined scheduling approach could be beneficial—one that accounts for both the execution duration of coroutines and the memory latency.

4 Projected Time-frame

The tutorial is intended as an interactive, hands-on tutorial with a duration of approximately 45 minutes. We plan to start with an introduction of the Linux perf subsystem and how perf-cpp interacts with it (roughly 10 minutes).

Following this, we will give a practical introduction to performance profiling, including a demonstration on how to set up the perf-cpp library (10 minutes).

Finally, the B-tree with optimistic lock coupling [LHN19] will serve as the poster child for a more complex data structure. We plan a live demonstration on how sampling can be used to, e.g., identify the effectiveness of software prefetching as detailed in Section 3.2. This part is explicitly intended for audience interaction, as we plan to provide a script (e.g., via *GitHub*), that will build and execute the benchmark on the devices of the participants. The results will be compared to highlight the differences that can occur on different types of hardware. We will also have some “backup” results from other machines for comparisons (25 minutes).

Acknowledgments

We thank the anonymous reviewers for their helpful feedback and suggestions. This work is funded by the DFG Priority Program *Disruptive Memory Technologies* (SPP2377) as part of the project *Memory Diplomat* (grant number 502384507) and by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr-Institute for Machine Learning and Artificial Intelligence*.

[2] We found that scheduling 12 coroutines simultaneously yielded the best performance on our system.

Bibliography

[AH17] Akiyama, Soramichi; Hirofuchi, Takahiro: Quantitative Evaluation of Intel PEBS Overhead for Online System-Noise Analysis. In: Proceedings of the 7th International Workshop on Runtime and Operating Systems for Supercomputers, ROSS@HPDC. ACM, pp. 3:1–3:8, 2017.

[AM24] AMD: AMD uProf. https://www.amd.com/de/developer/uprof.html, 2024. Accessed 21 August 2024.

[Be21] Beischl, Alexander; Kersten, Timo; Bandle, Maximilian; Giceva, Jana; Neumann, Thomas: Profiling dataflow systems on multiple abstraction levels. In: EuroSys '21: Sixteenth European Conference on Computer Systems. ACM, pp. 474–489, 2021.

[Co10] Cooper, Brian F.; Silberstein, Adam; Tam, Erwin; Ramakrishnan, Raghu; Sears, Russell: Benchmarking cloud serving systems with YCSB. In: Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC. ACM, pp. 143–154, 2010.

[Da23] Dang, Weizhen; Yu, Tao; Wang, Haibo; Li, Fenghua; Wang, Jilong et al.: PerfMon: Measuring application-level performance in a large-scale campus wireless network. China Communications, 20(3):316–335, 2023.

[DM10] De Melo, Arnaldo Carvalho: The new linux'perf'tools. In: Slides from Linux Kongress. volume 18, pp. 1–42, 2010.

[Dr07] Drongowski, Paul J: Instruction-based sampling: A new performance analysis technique for AMD family 10h processors. Advanced Micro Devices, 1(3):11, 2007.

[Dr10] Drongowski, Paul J.; Yu, Lei; Swehosky, Frank; Suthikulpanit, Suravee; Richter, Robert: Incorporating Instruction-Based Sampling into AMD CodeAnalyst. In: IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS. IEEE Computer Society, pp. 119–120, 2010.

[HLW20] He, Yongjun; Lu, Jiacheng; Wang, Tianzheng: CoroBase: Coroutine-Oriented Main-Memory Database Engine. Proc. VLDB Endow., 14(3):431–444, 2020.

[In24a] Intel®: Intel® 64 and IA-32 Architectures Software Developer's Manual. https://cdrdv2.intel.com/v1/dl/getContent/671200, 2024. Accessed 17 October 2024.

[In24b] Intel®: Intel® VTune™ Profiler. https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html, 2024. Accessed 21 August 2024.

[Jo18] Jonathan, Christopher; Minhas, Umar Farooq; Hunter, James; Levandoski, Justin J.; Nishanov, Gor V.: Exploiting Coroutines to Attack the "Killer Nanoseconds". Proc. VLDB Endow., 11(11):1702–1714, 2018.

[KMT24] Kühn, Roland; Mühlig, Jan; Teubner, Jens: How to Be Fast and Not Furious: Looking Under the Hood of CPU Cache Prefetching. In: Proceedings of the 20th International Workshop on Data Management on New Hardware, DaMoN. ACM, pp. 9:1–9:10, 2024.

[Kü23] Kühn, Roland; Biebert, Daniel; Hakert, Christian; Chen, Jian-Jia; Teubner, Jens: Towards Data-Based Cache Optimization of B+-Trees. In: Proceedings of the 19th International Workshop on Data Management on New Hardware, DaMoN. ACM, pp. 63–69, 2023.

[Le18] Leis, Viktor: PerfEvent. https://github.com/viktorleis/perfevent, 2018. Accessed 23 November 2024.

[LHN19] Leis, Viktor; Haubenschild, Michael; Neumann, Thomas: Optimistic Lock Coupling: A Scalable and Efficient General-Purpose Synchronization Method. IEEE Data Eng. Bull., 42(1):73–84, 2019.

[LLS13] Levandoski, Justin J.; Lomet, David B.; Sengupta, Sudipta: The Bw-Tree: A B-tree for new hardware platforms. In: 29th IEEE International Conference on Data Engineering, ICDE. IEEE Computer Society, pp. 302–313, 2013.

[Mi24] Miksits, Samuel; Shi, Ruimin; Gokhale, Maya B.; Wahlgren, Jacob; Schieffer, Gabin; Peng, Ivy: Multi-level Memory-Centric Profiling on ARM Processors with ARM SPE. CoRR, abs/2410.01514, 2024.

[MT21] Mühlig, Jan; Teubner, Jens: MxTasks: How to Make Efficient Synchronization and Prefetching Easy. In: SIGMOD: International Conference on Management of Data. ACM, pp. 1331–1344, 2021.

[Mü23] Mühlig, Jan: perf-cpp: Access Performance Counters from C++ Applications. https://github.com/jmuehlig/perf-cpp, 2023. Accessed 23 November 2024.

[No20] Noll, Stefan; Teubner, Jens; May, Norman; Böhm, Alexander: Analyzing memory accesses with modern processors. In: 16th International Workshop on Data Management on New Hardware, DaMoN. ACM, pp. 1:1–1:9, 2020.

[Ps17] Psaropoulos, Georgios; Legler, Thomas; May, Norman; Ailamaki, Anastasia: Interleaving with Coroutines: A Practical Approach for Robust Index Joins. Proc. VLDB Endow., 11(2):230–242, 2017.

[Ps19] Psaropoulos, Georgios; Legler, Thomas; May, Norman; Ailamaki, Anastasia: Interleaving with coroutines: a systematic and practical approach to hide memory latency in index joins. VLDB J., 28(4):451–471, 2019.

[Sa23] Sasongko, Muhammad Aditya; Chabbi, Milind; Kelly, Paul H. J.; Unat, Didem: Precise Event Sampling on AMD Versus Intel: Quantitative and Qualitative Comparison. IEEE Trans. Parallel Distributed Syst., 34(5):1594–1608, 2023.

[Sc23] Schubert, Nils L.; Grulich, Philipp M.; Zeuch, Steffen; Markl, Volker: Exploiting Access Pattern Characteristics for Join Reordering. In: Proceedings of the 19th International Workshop on Data Management on New Hardware, DaMoN. ACM, pp. 10–18, 2023.

[We16] Weaver, Vincent M: Advanced hardware profiling and sampling (PEBS, IBS, etc.): creating a new PAPI sampling interface. Technical Report UMAINE-VMWTR-PEBS-IBS-SAMPLING-2016-08. University of Maine, Tech. Rep., 2016.

[Yi20] Yi, Jifei; Dong, Benchao; Dong, Mingkai; Chen, Haibo: On the precision of precise event based sampling. In: APSys '20: 11th ACM SIGOPS Asia-Pacific Workshop on Systems. ACM, pp. 98–105, 2020.

[ZF15] Zeuch, Steffen; Freytag, Johann-Christoph: Selection on Modern CPUs. In: Proceedings of the 3rd VLDB Workshop on In-Memory Data Mangement and Analytics. ACM, pp. 5:1–5:8, 2015.

Unleashing the Intel Data Streaming Accelerator

André Berthold [1], Lennart Schmidt [1], Wolfgang Lehner [1], and Horst Schirmeier [1]

Abstract:

In-memory data processing is the state-of-the-art approach for large-scale data analytics. New developments in both memory technologies and application-specific accelerators introduce new opportunities and challenges to a computer system. To this end, Intel offers a new on-chip memory-operation accelerator called Data Streaming Accelerator (DSA) in modern Xeon Scalable processors, which enables higher data-transfer throughput by offloading memory operations from the CPU.

This tutorial introduces the DSA programming model and aims to enable participants to exploit the superior performance of the DSA over the CPU in terms of memory operations, e.g. to perform fast data relocation in a heterogeneous memory system.

Keywords: Hardware accelerator, Intel DSA, Intel DML, HBM2, memory management

1 Introduction

Fueled by falling DRAM prices and advances in memory technology, in-memory data processing has been popularized since the early 2000s [PZ12]. Affordable large memories made it for the first time possible to store whole databases in main memory, making data access and subsequent processing significantly more efficient than designs involving background storage at query time. Modern multi-tiered memory systems add the challenge that data must be moved from one memory tier to another to speed up a subsequent operation, or to free up space for another [Ge23; KCA21].

If such large-scale data transfers are executed by load-store loops on CPU cores, they compete with compute tasks, affecting end-to-end processing runtime. Following a long tradition of DMA controllers that allow offloading such tasks, as a remedy Intel introduced an on-chip DMA engine named *Data Streaming Accelerator (DSA)* in 4th generation Intel Xeon Scalable processors [Ku24]. The DSA is a high-performance data-copy and transformation accelerator that allows to offload simple memory operations such as DRAM-to-DRAM memory segment copy.

In this tutorial, we provide an overview of the DSA architecture, explain its programming model using on Intel's *Data Mover Library* (DML) based on concrete source-code examples, and demonstrate DSA setup and actual offloading benefits on a dual-socket server machine. The main objective of this tutorial is to provide attendees with an idea of how to set up and use the DSA to offload and speed up data transfers:

[1] Technische Universität Dresden, Dresden, Germany, andre.berthold@tu-dresden.de, https://orcid.org/0009-0007-2382-8053; lennart.schmidt1@tu-dresden.de, https://orcid.org/0009-0002-7979-0262; wolfgang.lehner@tu-dresden.de, https://orcid.org/0000-0001-8107-2775; horst.schirmeier@tu-dresden.de, https://orcid.org/0000-0002-1427-9343

doi:10.18420/BTW2025-139

- We briefly describe the architecture of the Intel DSA and its implemented operations,
- give instructions for setting up DSA engines on a system,
- introduce the high-level C++ and low-level C APIs provided by the DML for programming the DSA, and
- quantitatively compare the DSA's data throughput with a CPU-implemented baseline.

The tutorial comprises a 45-minute interactive presentation with live-demonstration elements. The target audience are developers and researchers from both industry and academia, in particular those who specialize in HPC, OLAP or memory management. We aim at a content level that is suitable for anyone not yet familiar with Intel DSA, and expect basic knowledge of the C/C++ programming language and the Linux command line.

2 Architecture of the Intel DSA

The Intel *Data Streaming Accelerator* is an on-chip accelerator for simple memory operations. A CPU can offload operations like copying memory regions, filling memory regions with a byte pattern, comparing memory regions, or even checksum calculation [Ku24]. The DSA works concurrently to the CPUs and thus frees up CPU cycles for other calculations. It was integrated in the 4th and 5th generation of Intel Xeon Scalable processors ("Sapphire Rapids" and "Emerald Rapids" architectures). Additionally, a new and improved version of the DSA, the *DSA 2.0*, is now available with the Intel Xeon 6 processors [In24b]. The DSA can be accessed via a PCIe-like interface, which allows communication with the necessary host system components [Ku24].

2.1 DSA Hardware Perspective

Fig. 1 shows a block diagram of the high-level hardware architecture of a DSA. To offload work to the DSA, the CPU pushes a *work descriptor* (WD) to a memory-mapped IO region called a *portal*. WDs contain all essential information for the task, like type of operation, pointer to the (to be written) completion record, and address and size of the memory buffers [Ku24].

Then, the WDs are placed in the *work queue* (WQ) that corresponds to the portal region. WQs are organized into user defined groups. Each group contains a number of WQs that compete for the *processing engines* (PEs) assigned to this group. It is noteworthy that a configuration with only one PE can reach peak throughput of a DSA device. However, configurations with multiple PEs might still make sense, as others PEs can work in parallel, while one PE stalls at a memory access or waits for the operating system to resolve a page fault. In Fig. 1, *Group 0* is configured to consist of *WQ 0* and *WQ 1* and one PE, namely *PE 0*. For every group an arbiter orchestrates the dispatching of the WDs at the head of the WQs to free PEs based on quality of service and fairness considerations [Ku24].

When a regular WD enters the PE, it proceeds directly to the PE's internal arbiter. However, as a batch descriptor conceptually combines multiple WDs and only contains information

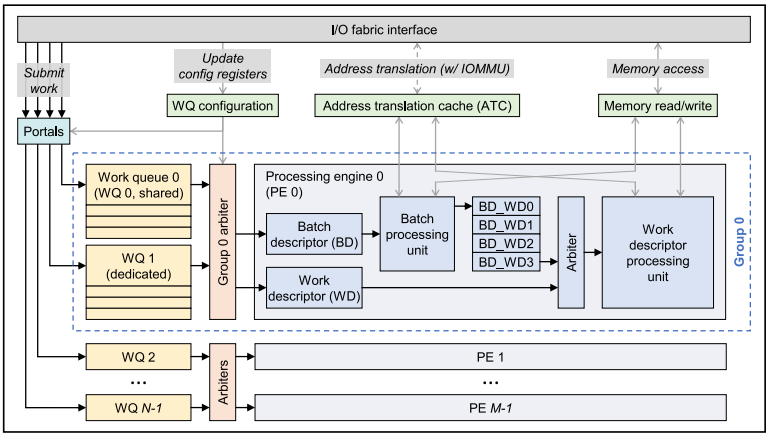


Fig. 1: Architectural overview of a DSA [Ku24]: Work descriptors are fetched from portals (gray) and stored in work queues (WQ, yellow), which, orchestrated by the priority decisions of the arbiter (red), feed processing engines (PE, blue).

where the actual work descriptors are stored, the batched regular WDs must be fetched from memory first. The PE's arbiter hands the WDs one by one to the work descriptor processing unit, which carries out the specified work by (i) translating virtual buffer addresses in the WD to physical ones, (ii) reading source data from memory, (iii) executing the specified operation, (iv) updating the destination buffer, and (v) writing the completion record or generating an interrupt if requested [Ku24].

3 Programming Model

3.1 DSA Configuration

Before the DSA is ready to receive WDs, it needs to be configured with the `accel-config` tool. It provides a command-line interface to change the configuration, and also a shared library for applications to query and modify the configuration programmatically [In24a].

To submit tasks to the Intel DSA and to fetch the corresponding completion records, we will rely on Intel's *Data Mover Library (DML)*, which was designed to simplify the usage of the accelerators. The DML offers a low- and a high-level C/C++ interface for interacting with the DSA.

When using the DML to submit tasks to the DSA, we first must prepare the data that should be used. Next, we describe the operation to be carried out. In the high-level C++ interface, this is merged with the actual task submission, where the information about the task is sent to the work queue of the DSA (if execution on the hardware was specified). To fetch the results, the library tries to fetch the completion record, which should be checked for error codes that indicate problems during execution of the task, before working with the data involved in the submitted task.

```
1  // memory regions for a DML operation have
2  // to be passed as dml::data_view objects
3  dml::data_view src_v = dml::make_view(src, size);
4  dml::data_view dst_v = dml::make_view(dst, size);
5
6  // asynchronous copy operation; path could be
7  // dml::software, dml::hardware, or dml::automatic
8  auto handler =
9      dml::submit<path>(dml::mem_copy, src_v, dst_v);
10
11 // ... do other asynchronous operation
12
13 // wait for the copy result
14 auto result = handler.get();
15 if (result.status != dml::dml::status_code::ok)
16     return -1;
```

Fig. 2: C++ Code snippet for an asynchronous copy operation using the high-level DML API.

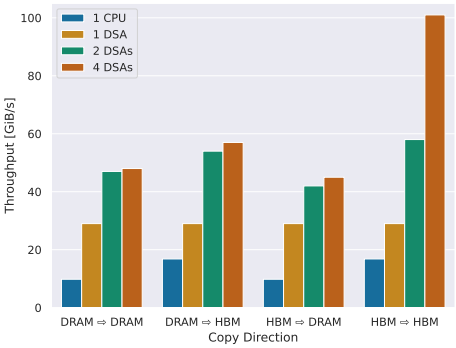


Fig. 3: Transfer comparison for up to 4 DSAs.

Fig. 2 shows a code snippet using the high-level DML interface for asynchronously executing a copy operation. This API operates on `dml::data_views`, created with `dml::make_view`. This function takes a pointer to the data and the number of elements (lines 3–4).

To execute a task, the API provides two functions `dml::execute` (synchronous) and `dml::submit` (asynchronous). Both require the execution path as template parameter, which specifies whether the operation should be executed on the CPU (`dml::software`), on the accelerator (`dml::hardware`), or if the executing device should be selected automatically (`dml::automatic`). The operation to be carried out is specified via an operation code and additional operation-specific parameters. In Fig. 2 we use `dml::mem_copy` as operation code and the mandatory source and destination data views (lines 8–9). A list of available operations and their required parameters can be found in the DML documentation [In22].

The synchronous interface (`dml::execute`) directly returns a status code that tells whether the operation was carried out successfully. The asynchronous interface (`dml::submit`), as used in Fig. 2, returns a `dml::handler`, a handle to the concurrently running task, which is used to await the task's termination using `dml::handler::get` (line 14) to receive a result object. To utilize the full potential of the Intel DSA it is advisable to use the asynchronous interface, to be able to carry out additional computations on the CPU while the DSA is busy. A working example of the code in Fig. 2 and more example code can be found in the tutorial repository [BS25].

4 Evaluation

4.1 System Setup

All the following results were measured on a dual-socket server equipped with two Intel Xeon Max 9468 CPUs. These CPUs feature 48 cores per socket. Each socket is furthermore

equipped with 64 GiB of HBM and 256 GiB of DDR5 DRAM. For each of the sockets, we activated the Sub-NUMA Clustering-4 (SNC4) mode, which splits each CPU and RAM into four logical NUMA nodes. The HBM is divided and exposed as four additional NUMA nodes. Using SNC4 in general should lead to a higher throughput and lower latency [In23].

We can logically group the NUMA nodes into *pair*s. Each pair consists of a HBM node with 16 GiB of memory and a DRAM node with 64 GiB of memory and 12 CPU cores. Additionally, every pair features a DSA, resulting in 4 DSAs per socket. Each of the benchmarks was run five times and the resulting throughput values were averaged.

4.2 Intra-Socket Data Transfer

Data locality is the center of attention for in-memory processing [Ki14]. In this paper we focus on intra-socket data transfers, based on prior work. For further details of inter-socket data transfers and the general application of these findings, please refer to Berthold et al. [Be24].

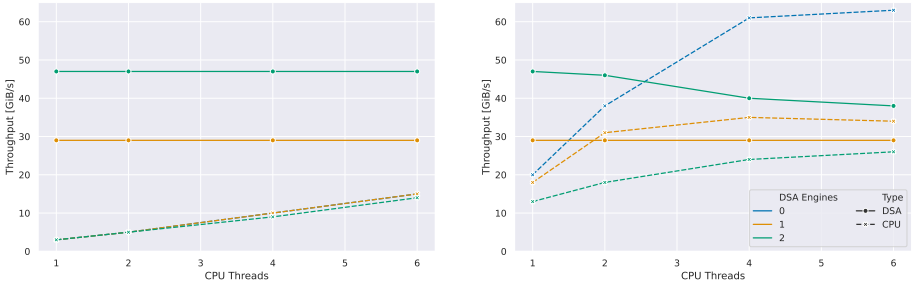
4.2.1 Data transfer

First, we examine the raw throughput we can achieve while transferring data. Therefore, we transfer 1 GiB of data from one pair to a different pair within the same socket.

For the CPU transfer we used the `memcpy` implementation of `glibc` 2.40. We decided not to use the software path option of the DML, as it performs poorly in terms of data throughput – probably because it was only meant as a backup implementation for the case when the accelerator fails (when `dml::automatic` is selected). For measuring the accelerator performance, we used the DSAs of one socket, leading to a maximum of 4 DSAs being used at once.

Fig. 3 shows the results of this experiment. In the comparison of one CPU core to one DSA, we see speedups of 1.73x–2.96x. Furthermore, a single DSA achieves a stable 29 GiB/s, regardless of the source or destination, while for the CPU the destination matters.

When using multiple DSAs, the throughput increases with the number of DSAs used. In the case of DRAM as source or destination, this throughput increases dramatically for 2 DSA, but stagnates beyond that, suggesting a saturation of the available bandwidth. The theoretical peak memory bandwidth of the system is around 65 GiB/s, because each DRAM node consists of two DDR5-4800 DIMMs. For HBM as source and destination, the throughput scales almost linearly up to 101 GiB/s. A multi-threaded CPU execution is slower by several factors and therefore not shown in the figure. A general observation is that for the transfer of larger data sets, the DSA is particularly beneficial, and for further increases in throughput multiple DSAs should be used [Be24].



(a) Concurrent compute-intensive operations CPU,

(b) Concurrent memory-intensive operations on CPU using AVX512.

Fig. 4: Concurrent execution of CPU and DSA: comparison for up to 4 DSAs and 6 CPU threads.

4.2.2 Concurrent execution

One of the selling points of the DSA is that it frees up CPUs that can be used to execute other operations in parallel. These operations can be categorized into compute- and memory-intensive operations.

As a representative for compute-intensive operations, we implemented a straightforward summation algorithm that calculates the sum of all integers within a specified memory region. For a memory-intensive operation, we use the vectorized version of that algorithm. The input data for this experiment resides in the local DRAM, and both operations are executed with up to six threads. This suffices to saturate the memory bus for the memory-intensive operation. Additionally, in this experiment the DSA concurrently transfers data from the local DRAM node to a different DRAM node.

As we can see in Fig. 4a, concurrent compute intensive operations do not influence the throughput of the DSA and vice versa. The results for a concurrent data transfer and a memory-intensive operation are shown in Fig. 4b. The highest observed throughput of the memory-intensive operation is around 63 GiB/s with 6 threads and no DSA transfer. A loss of throughput can be observed as soon as a concurrent data transfer occurs. For instance, with one DSA the throughput is, as seen before, a constant 29 GiB/s, even with 6 concurrent memory-intensive threads. However, the throughput of the memory-intensive operation with 6 threads is lower by this amount.

With two DSAs, the memory-intensive operation has some effect on the achieved throughput of the DSAs. Before, in Fig. 4a, the DSAs were able to achieve a constant 48 GiB/s. Now, with the concurrent memory-intensive operation, the DSAs start out with a throughput of 48 GiB/s and slows down to around 39 GiB/s with six concurrent threads. The benchmark's total throughput for six compute intensive threads stays at around 64 GiB/s with and without DSA.

Fig. 4b with one DSA shows a prioritization of the DSA over other operations. This

prioritization, however, is not a *total* prioritization, as the results of the experiment with two DSAs show. The limited bandwidth of the DRAM means that in a situation with a data transfer and a concurrent memory-intensive operation, some sort of consideration of priorities has to be done. We provided an example for this in prior work [Be24].

5 Tutorial Organization

5.1 Schedule

I – Introduction (5 min): We will highlight the relevance of Intel DSA for accelerating data-intensive workloads to make a case for the participants to use the DSA in their own software.

II – Intel DSA architecture (10 min): We dive into the hardware architecture and explain the most important building parts of the DSA, give an overview of the available operations of the DSA, and outline DSA's memory access path, work queues, and ordering guarantees. Thus, we provide insights into the internal structure of the DSA to allow a better understanding of the setup and programming methodology explained later in the tutorial.

III – Intel DSA configuration (7 min): We outline the different parameters available to configure the engines, work queues, and groups of an Intel DSA device and explain their effect, and we show how to apply a certain configuration. So, we provide the attendees with an understanding of how the configuration of the DSA changes its behavior and how they can set up a DSA work queue (which is the entry point for work descriptors).

IV – Intel DSA Programming (13 min): We introduce the DML, which provides a low-level C API and a high-level C++ API to offload work to the DSA (more specifically the DSA work queue). Additionally, we explain the interface and the effect of the functions and methods provided. We provide a demo where we will compile, and execute an example application that tasks the DSA with a data transfer. We equip the attendees with the necessary know-how to actually submit tasks to the DSA using Intel DML. We exemplarily show code that submits an asynchronous data-transfer task to a DSA work queue.

V – Performance Comparison with CPU (10 min): We will present benchmark results published by Berthold et al. [Be24], which show the throughput characteristics of DSA compared to a general purpose CPU on our research system (Intel Xeon MAX 9468) for different data placement scenarios. Based on these benchmarks, we can draw conclusions about when to use or not to use the DSA. We outline the DSA's actual performance and give recommendations when to use or not to use the DSA.

5.2 Tutorial Contributor

André Berthold – Primary Presenter

Affiliation	Technische Universität Dresden, Chair of Operating Systems
Job title	Research Assistant
Postal address	Nöthnitzer Str. 46, 01187 Dresden, Germany
Email address	andre.berthold@tu-dresden.de

Lennart Schmidt – Presenter

Affiliation	Technische Universität Dresden, Dresden Database Research Group
Job title	Research Assistant
Postal address	Nöthnitzer Str. 46, 01187 Dresden, Germany
Email address	lennart.schmidt1@tu-dresden.de

Wolfgang Lehner – Supporter

Affiliation	Technische Universität Dresden, Dresden Database Research Group
Job title	Full Professor
Postal address	Nöthnitzer Str. 46, 01187 Dresden, Germany
Email address	wolfgang.lehner@tu-dresden.de

Horst Schirmeier – Supporter

Affiliation	Technische Universität Dresden, Chair of Operating Systems
Job title	Full Professor
Postal address	Nöthnitzer Str. 46, 01187 Dresden, Germany
Email address	horst.schirmeier@tu-dresden.de

6 Conclusion

This tutorial enables the attendees to take their first own steps with the Intel *Data Streaming Accelerator*. After an introductory architecture overview and a description of the DSA's hardware interface, the presenters introduce the programming model based on concrete examples with Intel's *Data Mover Library*. Evaluation results from a dual-socket Intel Xeon server demonstrate that the DSA not only offloads large-scale memory operations from the general-purpose CPUs, but also reaches higher throughput rates when *High Bandwidth Memory* is involved.

Acknowledgments This work was partly supported by the German Research Foundation (DFG) priority program SPP 2377 under grants no. SCHI 1447/1-1 and LE 1416/30-1.

References

[Be24] Berthold, A.; Fürst, C.; Obersteiner, A.; Schmidt, L.; Habich, D.; Lehner, W.; Schirmeier, H.: Demystifying Intel Data Streaming Accelerator for In-Memory Data Processing. In: Proceedings of the 2nd Workshop on Disruptive Memory Systems. DIMES '24, Association for Computing Machinery, Austin, TX, USA, pp. 9–16, 2024, ISBN: 9798400713033, DOI: 10.1145/3698783.3699383.

[BS25] Berthold, A.; Schmidt, L.: noDMC-DSA-Tutorial, https://github.com/adbexy/noDMC-DSA-Tutorial, 2025.

[Ge23] Geyer, A.; Pietrzyk, J.; Krause, A.; Habich, D.; Lehner, W.; Färber, C.; Willhalm, T.: Near to Far: An Evaluation of Disaggregated Memory for In-Memory Data Processing. In: 1st Workshop on Disruptive Memory Systems (DIMES@SOSP '23). Pp. 16–22, 2023.

[In22] Intel Corporation: Intel® DML 1.2.0 documentation, 2022, URL: https://intel.github.io/DML/index.html#, visited on: 02/07/2025.

[In23] Intel: Intel Xeon CPU Max Series – Configuration and Tuning Guide, English, version Revision 003, Intel, 2023, URL: https://www.intel.com/content/www/us/en/content-details/787743/intel-xeon-cpu-max-series-configuration-and-tuning-guide.html, visited on: 02/12/2025.

[In24a] Intel: Intel® Data Streaming Accelerator User Guide, English, version Revision 003, Intel, 2024, URL: https://www.intel.com/content/www/us/en/content-details/759709/intel-data-streaming-accelerator-user-guide.html, visited on: 02/12/2025.

[In24b] Intel Corporation: Move Data Faster for Better Business Insights with Intel DSA, 2024, URL: https://www.intel.com/content/www/us/en/content-details/787923/move-data-faster-for-better-business-insights-with-intel-dsa.html, visited on: 01/25/2025.

[KCA21] Kim, J.; Choe, W.; Ahn, J.: Exploring the design space of page management for multi-tiered memory systems. In: 2021 USENIX Annual Technical Conference (USENIX ATC 21). Pp. 715–728, 2021.

[Ki14] Kiefer, T.; Kissinger, T.; Schlegel, B.; Habich, D.; Molka, D.; Lehner, W.: ERIS live: a NUMA-aware in-memory storage engine for tera-scale multiprocessor systems. In (Dyreson, C. E.; Li, F.; Özsu, M. T., eds.): International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014. Pp. 689–692, 2014.

[Ku24] Kuper, R.; Jeong, I.; Yuan, Y.; Wang, R.; Ranganathan, N.; Rao, N.; Hu, J.; Kumar, S.; Lantz, P.; Kim, N. S.: A Quantitative Analysis and Guidelines of Data Streaming Accelerator in Modern Intel Xeon Scalable Processors. In: ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24). Pp. 37–54, 2024.

[PZ12] Plattner, H.; Zeier, A.: In-Memory Data Management An Inflection Point for Enterprise Applications. Springer Science & Business Media, 2012.

Feasibility Analysis of Semi-Permanent Database Offloading to UPMEM Near-Memory Computing Modules

Birte Friesel [1], Marcel Lütke Dreimann [1], and Olaf Spinczyk [1]

Abstract: While near-memory computing offers significant speedups for parallel database workloads, the only currently available commercial implementation, UPMEM PIM, suffers from setup and data transfer overheads. Due to these, its past applications in database systems have focused on analytical (read-only) workloads with data permanently residing in UPMEM modules, and dynamic partial offloading of queries that takes these overheads into account. Here, we examine a middle ground: letting data reside in UPMEM modules across consecutive queries, and evicting it only when it is altered by a write statement or when another data set should be offloaded instead. Specifically, we combine benchmark-based performance models with a task-based scheduling simulator to determine at which point offloading to UPMEM is worth the incurred overheads for a variable number of available CPU cores, UPMEM ranks, table sizes, and consecutive UPMEM-compatible database queries. We show that this approach leads to reliable latency simulations that can be verified with real-world benchmarks. Using these simulations, we find that offloading individual columns to UPMEM becomes worthwhile once table size exceeds 2^{22} to 2^{27} rows, with a small number of consecutive queries (one to several dozen) needed to amortize the setup and data transfer overheads.

Keywords: Near-Memory Computing, UPMEM, Databases, Simulation

1 Introduction

The past decades have seen steady improvements in CPU performance and memory capacity, whereas memory access latencies have remained largely constant [LRJ18]. Although parallel algorithms can speed up database operations, they are still affected by the so-called *memory wall*: the interface between CPU and DRAM has become a bottleneck [Mc04].

The concept of *near-memory computing* (NMC) promises a way out of this. By providing additional compute cores close to data in main memory (i.e., next to or built into individual memory chips), data-intensive operations can directly access their respective memory regions without having to compete for CPU capacity and memory controller bandwidth. UPMEM PIM, which is the only commercially available implementation of NMC at the time of writing [Gó22], achieves this by embedding *DRAM Processing Units* (DPUs) into DDR4 memory modules.

Due to design limitations of UPMEM PIM and the DDR memory interface, data residing on UPMEM PIM modules cannot be directly accessed by the CPU, but must be transferred

[1] Universität Osnabrück, AG Eingebettete Softwaresysteme, Wachsbleiche 27, 49090 Osnabrück, Deutschland,
birte.friesel@uos.de, https://orcid.org/0000-0002-0688-9440;
marcel.luetkedreimann@uos.de, https://orcid.org/0009-0007-2426-4798;
olaf.spinczyk@uos.de, https://orcid.org/0000-0001-9469-2367

doi:10.18420/BTW2025-140

between DRAM and UPMEM PIM whenever switching between CPU and DPU execution. Moreover, each DPU only has access to a single 64 MiB partition of data, with no shared memory across DPUs. Altogether, this makes UPMEM PIM behave more like an accelerator than a true NMC implementation [FLS23].

Related works have already shown that UPMEM PIM offers significant speedups for highly parallel, analytical database workloads such as table scans, joins, or graph database lookups [BJS23a; BJS23c; Li23]. However, these have focused on large databases and complex queries, where the massive parallelism of DPU processing amortizes the overhead of DPU allocation and data transfer between DRAM and UPMEM. In this paper, we look at a broader configuration space for less complex analytical query kernels, and determine the precise break-even points between CPU and DPU execution for a variety of CPU and DPU allocations, table sizes, and database workloads.

We first present a method for combining performance models with a task-based scheduling simulator to predict the makespan (i.e., total execution time) of a series of analytical database operations for variable CPU- and UPMEM-based execution settings. Using this, we determine the break-even points at which DPU performance (including overheads) exceeds CPU performance for a variety of CPU core allocations, DPU allocations, and column sizes. By validating a subset of break-even points on real hardware, we show that our simulation-based state space exploration allows for accurate offloading decisions.

The remainder of this paper is organized as follows. After a brief introduction to UPMEM PIM in Sect. 2, Sect. 3 describes our methodology and evaluation target. Sect. 4 discusses the corresponding findings and validates the simulation results on real hardware. Sect. 5 examines related work, and Sect. 6 concludes the paper.

2 UPMEM PIM

UPMEM PIM is the first and, at the time of writing, still only commercially available NMC implementation. It builds upon the DDR4 memory interface, which makes it compatible with a wide range of server platforms – the only requirement is a suitably patched BIOS. Each UPMEM module provides 8 GiB of memory, divided into two ranks of 4 GiB. Each rank is equipped with 64 DRAM processing units (DPUs): 32-bit RISC cores running at approx. 400 MHz, depending on hardware revision. DPUs implement single program multiple data (SPMD) hardware multi-tasking with support for up to 24 tasklets. At least eleven tasklets are required for optimal performance [Gó22].

Each DPU has access to a distinct 64 MiB range of memory and cannot read or write any data outside of this memory area; synchronization across DPUs or ranks must happen via the CPU. Effectively, this partitions the 4 GiB of memory provided by an UPMEM rank into 64 distinct compute regions of 64 MiB each. Synchronization between tasklets running on the same DPU is supported.

Workloads that require little or no synchronization, such as table scans or vector math operations, can benefit from the immense parallelism offered by UPMEM [Gó22]. For instance, a parallel vector addition across more than 2,000 DPUs is orders of magnitude faster than the same vector addition performed by four CPU cores.

Due to the interleaving employed in UPMEM DIMMs, CPUs and DPUs do not share the same view on memory contents: a consecutive sequence of eight bytes is spread across the memory partitions of eight different DPUs. Hence, UPMEM memory is not available as conventional main memory. It must be managed explicitly via the UPMEM SDK, which uses a pool of helper threads to take care of the necessary data transformations. In order to use UPMEM's NMC capabilities, an application has to allocate a set of DPUs or ranks, upload a program to the allocated DPUs, and upload any data that the program needs. Once the computation is done, the application can retrieve output data and release the DPUs, upload a different program to work on the same data, or upload new data for the already-present program [Gó22]. Altogether, this makes UPMEM behave like an accelerator, and computational offloading implementations must take the setup costs and data transfer overheads associated with it into account [FLS23]. For a detailed introduction to UPMEM PIM and its relation to database workloads, we refer to related work [BJS23c; Gó22; Li23].

3 Methodology

Our configuration space exploration starts with micro-benchmarks of UPMEM SDK throughput and latency as well as query kernel latency for a variety of resource allocations and column sizes. After transforming those into performance models, we use a task-based scheduling simulator to determine CPU- and UPMEM-based makespans for a variable number of consecutive analytical queries on a table column. These simulations provide an exhaustive configuration space exploration, thus allowing us to determine the break-even points in Sect. 4. We now briefly describe our micro-benchmarks, the performance model learning process, and the scheduling simulator integration.

3.1 Micro-Benchmarks

We utilize an extended version of the CPU-DPU micro-benchmark provided by the PrIM benchmark suite to determine the latency of DPU allocation (`dpu_alloc`), program upload (`dpu_load`), and data transfer (`dpu_push_xfer`) between DRAM and UPMEM PIM modules [FLS24]. All benchmarks run on a server outfitted with two Intel Xeon Silver 4215 CPUs (16 cores / 32 threads across two sockets), 128 GiB DDR4-3200 DRAM (one 64 GiB module per socket), and 160 GiB UPMEM PIM (10 DIMMs / 20 ranks and thus 80 GiB per socket, using a DPU clock speed of 350 MHz). 2,543 of its 2,560 DPUs are usable for NMC, with the remaining 17 marked as defective. This is within the manufacturer-specified tolerance of up to 2.5 % defective DPUs [FLS23].

The machine has two NUMA nodes (one per socket). While NUMA configuration has little influence on UPMEM performance [FLS24], it is beneficial to place CPU tasks and data within the same node. Hence, we perform all benchmarks on NUMA node 1, using UPMEM SDK version 2024.2.0.

Our benchmarks rely on custom CPU and UPMEM implementations for COUNT and SELECT operators, reading a single column that holds 64-bit integer values. We assume a database systems that utilizes a column store memory layout [Fa17], and store the column as a dense array in main memory.

The operators accept a predicate $\bowtie \in \{<, \leq, =, \neq, \geq, >, \mathit{bit\_set}, \mathit{bit\_clear}\}$ and an argument a and determine the elements x with $x \bowtie a$. For COUNT, each CPU or DPU counts separately, with a single host CPU reducing the results to determine the count across the entire column. SELECT returns a bitmask that indicates the selected elements; rows are partitioned across CPUs or DPUs so that no post-processing is required. In both cases, we partition data for UPMEM execution so that it is spread evenly across the allocated DPUs, and use 16 tasklets per DPU. Our UPMEM implementation consists of a single binary that provides both kernels, hence the host must provide a 1-Byte kernel identifier in addition to predicate and argument in order to offload an operation to UPMEM.

Note that SELECT differs from the SEL benchmark provided by the PrIM benchmark suite [Gó22]. PrIM SEL returns the selected elements rather than bitmasks. This mandates a slow sequential data transfer process [FLS23].

We perform micro-benchmarks for allocation and program transfer latency involving 1 to 40 UPMEM ranks. UPMEM data transfer benchmarks use 8 B per DPU to determine transfer latency and 8 MiB per DPU to determine transfer throughput. Kernel latency benchmarks utilize 1 to 8 CPU cores, 1 to 40 ranks, 2^{20} to $2^{33} - 2^{30}$ rows (8 MiB to 56 GiB of column data), and a constant sequence of COUNT and SELECT queries with different predicates.

3.2 Performance Models

Kernel	COUNT		SELECT	
CPU (1C)	$4.5\,\text{ns} + 2.2\,\text{ns} \cdot \#\mathit{rows}$	±3.8 %	$0.0\,\text{ns} + 2.0\,\text{ns} \cdot \#\mathit{rows}$	±2.6 %
CPU (2C)	$5.2\,\text{ns} + 1.1\,\text{ns} \cdot \#\mathit{rows}$	±4.6 %	$5.2\,\text{ns} + 1.0\,\text{ns} \cdot \#\mathit{rows}$	±4.3 %
CPU (4C)	$0.9\,\text{ns} + 0.6\,\text{ns} \cdot \#\mathit{rows}$	±5.1 %	$0.0\,\text{ns} + 0.6\,\text{ns} \cdot \#\mathit{rows}$	±6.7 %
CPU (8C)	$0.0\,\text{ns} + 0.5\,\text{ns} \cdot \#\mathit{rows}$	±8.3 %	$4.8\,\text{ns} + 0.6\,\text{ns} \cdot \#\mathit{rows}$	±10 %
DPU	$242\,\mu\text{s} + 34\,\text{ns} \cdot \frac{\#\mathit{rows}}{\#\mathit{DPUs}}$	±12 %	$203\,\mu\text{s} + 42\,\text{ns} \cdot \frac{\#\mathit{rows}}{\#\mathit{DPUs}}$	±5.8 %

Tab. 1: Performance models and cross-validated prediction error for 1-/2-/4-/8-threaded CPU execution and UPMEM execution of COUNT and SELECT kernels.

We utilize our *regression model tree* (RMT) machine learning algorithm to transform benchmark results into performance models for DPU allocation, program transfer, data

transfer, and kernel execution time [FS22]. Tab. 1 shows an excerpt of the resulting models for kernel latency prediction and their cross-validated prediction error. UPMEM kernel latency scales linearly with the number of allocated DPUs, hence we provide a single model for all configurations. CPU kernel latency scales linearly for up to four cores, with additional cores giving diminishing returns. The underlying measurement uncertainty is less than 8 % for DPU execution and less than 4 % for CPU execution.

Setup cost and data transfer models are as follows. *#instr* refers to the number of instructions in the DPU program.

$$\begin{aligned}
\texttt{dpu_alloc}[ms] &= 13.5 + 12.3 \cdot \#ranks \\
\texttt{dpu_load}[\mu s] &= 820 + 4.1 \cdot \#instr \\
\text{write}[\mu s] &= 130 + 18 \cdot \min(\#ranks, 36) \\
\text{write}[GB/s] &= 11.3 \\
\text{read}[\mu s] &= 171 + 13 \cdot \min(\#ranks, 38) \\
\text{read}[GB/s] &= 2.9 + 0.1 \cdot \min(\#ranks, 28)
\end{aligned}$$

3.3 Query Simulations

In order to determine the break-even points between CPU and DPU execution, we examine scenarios with 1 to 100 consecutive COUNT or SELECT queries on random predicates, using a single DPU program that supports all evaluated operators and predicates. For UPMEM execution, each scenario allocates a set of ranks, uploads the DPU program, transfers the column, and then runs all queries on UPMEM. For CPU execution, all queries run on the CPU, with no setup costs or data transfer overheads.

Our configuration space consists of 200 different query scenarios (100 for COUNT and 100 for SELECT), 16 column sizes, 8 CPU core configurations, and 40 UPMEM rank configurations, leading to a total of 1,024,000 configurations. With each benchmark taking up to several seconds, an exhaustive benchmark-based exploration of this state space would take several weeks at least, and possibly even months. Hence, our analysis instead relies on *HetSim*, a simulator for task-based scheduling on heterogeneous hardware [LFS24].

For each configuration, we define a task set that executes the configured queries sequentially, and a system configuration that provides a certain number of CPU cores and UPMEM ranks. We augment task set and system configuration with automatically generated RMT performance models. These are similar to those shown in Tab. 1, but tailored towards the configured number of table rows, with variable CPU/DPU allocation. The task set includes a data object whose size corresponds to the number of 64-bit column elements as well as data objects that describe the input and output parameters of each query (see Sect. 3.1).

For UPMEM execution, HetSim automatically includes setup costs and the overhead of data transfer between DRAM and UPMEM memory. For each scenario, the column itself is transferred to UPMEM only once; query-specific input and output data is transferred for each query. A task set for n consecutive queries encodes the following instructions: `dpu_alloc_ranks`; `dpu_load`; `dpu_push_xfer` (write column data); $n \times$ { `dpu_push_xfer` (write kernel arguments); `dpu_launch` (run kernel); `dpu_push_xfer` (read kernel output) }. The CPU task set simply encodes $n \times$ `run_kernel`.

4 Findings

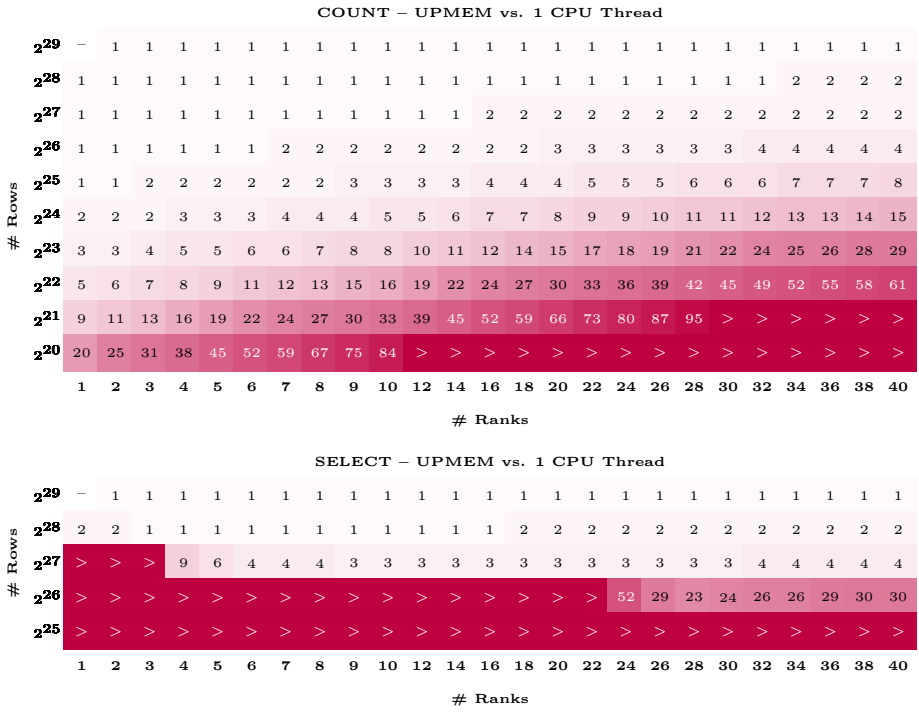


Fig. 1: Minimum number of consecutive COUNT and SELECT queries on 64-bit integer values needed to amortize the overhead of UPMEM over single-threaded CPU execution. The overhead is amortized if and only if the simulated makespan of UPMEM execution, including setup costs and data transfer overheads, is lower than the makespan of CPU execution. “>” indicates more than 99 queries; “–” indicates that the column is too large to fit into the specified number of ranks.

We are interested in determining at which point the speedup gained by UPMEM execution outweighs its setup costs and data transfer overheads. To this end, Fig. 1 and 2 show the minimum number of consecutive operations required to amortize the overhead of offloading query processing to UPMEM for a variable number of table rows and UPMEM ranks,

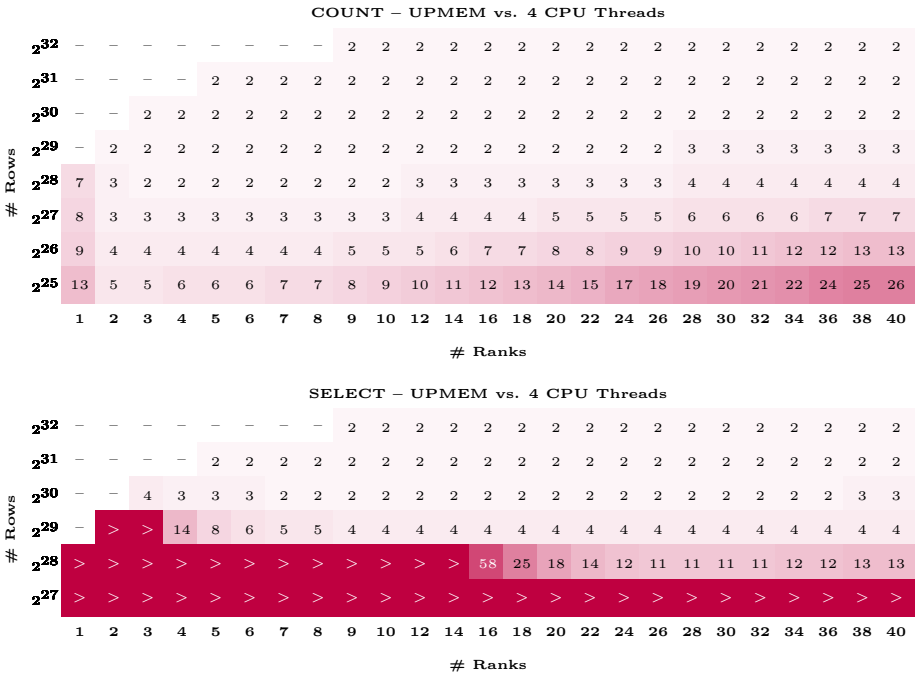


Fig. 2: Minimum number of consecutive COUNT and SELECT queries on 64-bit integer values needed to amortize the overhead of UPMEM over multi-threaded CPU execution. The overhead is amortized if and only if the simulated makespan of UPMEM execution, including setup costs and data transfer overheads, is lower than the makespan of CPU execution. ">" indicates more than 99 queries; "–" indicates that the column is too large to fit into the specified number of ranks.

compared against CPU configurations with one and four threads. A "1" cell indicates that offloading to UPMEM is beneficial even for just a single query, and an $n > 1$ indicates that n consecutive queries on the same column are needed to amortize the offloading overhead.

First off, we see that the break-even points of COUNT and SELECT differ by several powers of two. COUNT combines a simple kernel (see Tab. 1) with a minimal amount of input and output data, and thus benefits from UPMEM execution for relatively small tables: it is more efficient than single-threaded CPU execution for 2^{25} rows (256 MiB of data), and with just 2^{21} rows (16 MiB), less than ten consecutive queries are needed for overhead amortization. Meanwhile, SELECT is more complex and less UPMEM-suitable: its CPU implementation is as fast as COUNT, whereas its UPMEM implementation is 25 % slower. It also comes with higher data transfer overheads, as its output bitmask takes up $\frac{\#rows}{8}$ B that must be transferred back to DRAM. So, for SELECT, 2^{28} rows (2 GiB of data) are needed to be more efficient than single-threaded CPU execution with a single consecutive query.

Conversely, our results show that using UPMEM with SDK 2024.2.0 for small tables (less than 2^{21} rows / 16 MiB of data) is generally not worthy of consideration – below this point, COUNT needs dozens of consecutive queries to amortize UPMEM's setup overheads, and we cannot think of an even simpler and thus more UPMEM-friendly kernel.

When it comes to multi-threaded CPU execution (Fig. 2), we did not find configurations where a single consecutive query is sufficient to warrant offloading. Both COUNT and SELECT need two consecutive queries on at least 2^{28} (COUNT) and 2^{30} (SELECT) rows for UPMEM to be more efficient.

We also see that there is no best practice for allocating a certain number of ranks. Intuitively, one might expect that more ranks (and thus higher parallelism) are always beneficial. However, for both kernels, increasing the number of ranks reduces offloading performance in almost all cases. This is due to two opposing effects in UPMEM overheads. On the one hand, spreading the same amount of data across more ranks reduces kernel latency thanks to higher parallelism, while data transfer overhead for a constant amount of data is largely independent of the number of ranks. On the other hand, DPU allocation latency scales linearly with the number of DPUs, likely due to a sequential implementation in the SDK.

For instance, when comparing COUNT on UPMEM against multi-threaded CPU execution, the optimal configuration for 2^{28} rows is 3 to 10 ranks, whereas 2 or 3 ranks are the best option for 2^{25} rows. Meanwhile, optimal configurations for SELECT include 3 to 16 ranks (2^{28} rows, one CPU thread) and 9 to 40 ranks (2^{29} rows, four CPU threads). Overall, this shows that fine-grained performance models that consider setup costs, data transfer overheads, and kernel latencies are mandatory for making sensible offloading decisions.

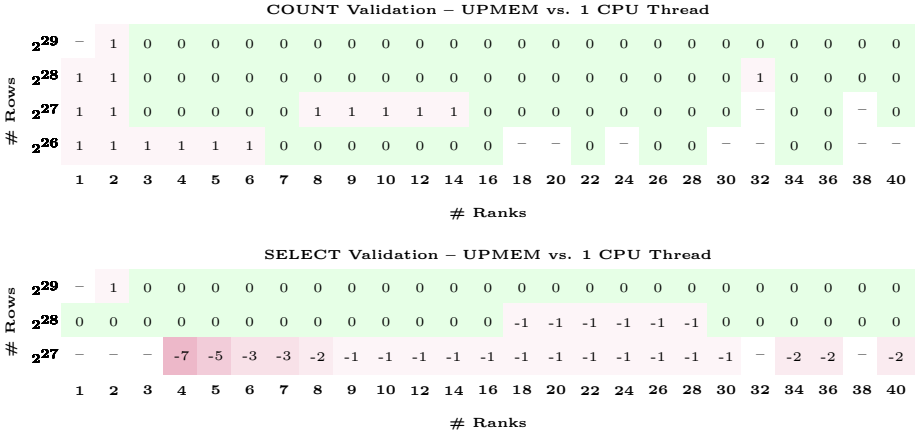


Fig. 3: Difference between break-even points determined by validation measurements and simulation results. Positive values indicate that the estimated break-even point at which UPMEM outperforms CPU execution was too low (i.e., the simulator over-estimated UPMEM performance), negative values indicate the inverse. "–" indicates that no validation data is available.

We validated a subset of the break-even points shown in Fig. 1 and 2 with benchmarks on real hardware, and found that the majority of results agree with them. For up to two ranks, our performance models under-estimate UPMEM setup costs, and predict a break-even point of one consecutive query where two are needed. When predicted UPMEM and CPU latency are within 10 % of each other, the predicted break-even point may also be off by one in either direction. With 8 % UPMEM measurement uncertainty, we consider these results to be acceptable. Fig. 3 shows an excerpt of the relation between our validation measurements and the simulation results reported above.

5 Related Work

Past publications on applying near-memory computing towards databases and related workloads broadly fall into two categories: NMC implementations that rely on simulators or custom FPGA builds [Co21; Le20], and UPMEM PIM. Here, we only cover the latter.

Baumstark et al. examine the benefits of using UPMEM PIM for analytical graph database queries on data that is permanently residing on UPMEM modules, using variable CPU and DPU allocations and table sizes [BJS23a]. When leaving out setup costs and data transfer overheads, they find that offloading to as little as 32 DPUs is beneficial, with more (up to 510) DPUs always causing a further reduction in execution time. This is consistent with our own findings (see Tab. 1). In their scenario, data transfer is only required once, at database startup. Their analysis shows that transfer overhead scales linearly with table size and is largely independent of the number of DPUs, which also agrees with our own findings.

Baumstark et al. also present a cost model for (partial) offloading of query operators to UPMEM, again in a graph database setting [BJS23c]. Here, the cost model includes data transfer, table size, number of DPUs and number of tasklets, but assumes equal read and write rates and leaves out setup costs. They find that, while UPMEM is still generally beneficial, its effectiveness depends on the selectivity of the executed query.

Lim et al. propose and evaluate an UPMEM-capable JOIN implementation, observing a 1.14× to 2× speedup over CPU execution [Li23]. However, their evaluation is limited to a single UPMEM and CPU core configuration.

Lastly, Baumstark et al. present a method for adaptive query compilation for data that is permanently residing in UPMEM modules [BJS23b]. By starting a query on the CPU and then switching to DPU execution once the query program has been compiled and uploaded, they are able to support nearly arbitrarily complex queries. We expect that our simulation-based approach is compatible with this method by extending the task sets with appropriate `dpu_load` calls, thus eliminating the need to provide a single DPU binary that implements all relevant queries and predicates.

While we see that cost models are already being used to drive offloading decisions, we are not aware of any approach that combines variable table sizes, CPU allocations, and

DPU allocations while also taking the complexity of the offloaded operators into account. Hence, we consider the simulation-based method outlined in this paper to provide a helpful perspective for further improvements of UPMEM-based offloading algorithms.

6 Conclusion

We have presented an approach for combining micro-benchmarks with simulated database operation sequences and used it to examine the break-even points between CPU and NMC execution (implemented using UPMEM PIM) of COUNT and SELECT operators for variable column sizes, CPU allocations, and DPU allocations. Our approach includes all UPMEM-specific setup costs and data transfer overheads, and is able to perform an exhaustive state space exploration with hours of simulation time rather than months of benchmarks. Artifacts are available at https://ess.cs.uos.de/git/artifacts/nodmc25-upmem-db.

Using these simulations, we have found that offloading individual analytical queries to UPMEM modules is only beneficial for sufficiently large tables, with 2^{25} to 2^{28} rows needed for UPMEM to be faster than CPU execution. The number of allocated UPMEM ranks also plays a role; its influence depends on operator complexity. We have found that there is no suitable heuristic for selecting an appropriate number of ranks: in general, neither the minimum number of ranks needed to fit all data nor the maximum number of ranks available in the system are most efficient. Instead, offloading decisions should be based on performance models that allow for finding the sweet spot between UPMEM setup costs (favouring as few ranks as possible) and kernel latency (favouring as many ranks as possible).

We have confirmed the accuracy of our simulation results by validating a subset of offloading decisions on real hardware, and found that the majority of break-even points obtained by simulation agree with real-world benchmark results. Most disagreements are off-by-one errors (i.e., the number of consecutive queries determined as the break-even point is too high or too low by one query) that occur when CPU and UPMEM latency are within the measurement uncertainty of the underlying benchmark and model data.

Altogether, this shows the benefit of cost models and thorough, simulation-based state space explorations using tools such as HetSim. We note that this method is not limited to UPMEM or NMC in general. It can also be used, for instance, to determine whether it is beneficial to transfer a database column to high-bandwidth memory prior to running a set of data-intensive operations on it.

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 502565817.

References

[BJS23a] Baumstark, A.; Jibril, M. A.; Sattler, K.-U.: Accelerating Large Table Scan using Processing-In-Memory Technology. In: BTW 2023. Gesellschaft für Informatik e.V., pp. 797–814, 2023, DOI: 10.18420/BTW2023-51.

[BJS23b] Baumstark, A.; Jibril, M. A.; Sattler, K.-U.: Adaptive Query Compilation with Processing-in-Memory. In: Proceedings of the 39th International Conference on Data Engineering Workshops. ICDEW '23, pp. 191–197, 2023, DOI: 10.1109/ICDEW58674.2023.00035.

[BJS23c] Baumstark, A.; Jibril, M. A.; Sattler, K.-U.: Processing-in-Memory for Databases: Query Processing and Data Transfer. In: Proceedings of the 19th International Workshop on Data Management on New Hardware. DaMoN '23, Association for Computing Machinery, pp. 107–111, 2023, DOI: 10.1145/3592980.3595323.

[Co21] Corda, S.; Kumaraswamy, M.; Awan, A. J.; Jordans, R.; Kumar, A.; Corporaal, H.: NMPO: Near-Memory Computing Profiling and Offloading. In: Proceedings of the 24th Euromicro Conference on Digital System Design. DSD '21, IEEE, pp. 259–267, 2021, DOI: 10.1109/DSD53832.2021.00048.

[Fa17] Faerber, F.; Kemper, A.; Larson, P.-Å.; Levandoski, J.; Neumann, T.; Pavlo, A.: Main Memory Database Systems. Foundations and Trends in Databases 8 (1-2), pp. 1–130, 2017, ISSN: 1931-7883, DOI: 10.1561/1900000058.

[FLS23] Friesel, B.; Lütke Dreimann, M.; Spinczyk, O.: A Full-System Perspective on UPMEM Performance. In: Proceedings of the 1st Workshop on Disruptive Memory Systems. DIMES '23, Association for Computing Machinery, pp. 1–7, 2023, DOI: 10.1145/3609308.3625266.

[FLS24] Friesel, B.; Lütke Dreimann, M.; Spinczyk, O.: Performance Models for Task-based Scheduling with Disruptive Memory Technologies. In: Proceedings of the 2nd Workshop on Disruptive Memory Systems. DIMES '24, Association for Computing Machinery, pp. 1–8, 2024, DOI: 10.1145/3698783.3699376.

[FS22] Friesel, B.; Spinczyk, O.: Regression Model Trees: Compact Energy Models for Complex IoT Devices. In: Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things. CPS-IoTBench '22, IEEE, pp. 1–6, 2022, DOI: 10.1109/CPS-IoTBench56135.2022.00007.

[Gó22] Gómez-Luna, J.; Hajj, I. E.; Fernandez, I.; Giannoula, C.; Oliveira, G. F.; Mutlu, O.: Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System. IEEE Access 10, pp. 52565–52608, 2022, DOI: 10.1109/ACCESS.2022.3174101.

[Le20] Lee, D.; Chang, A.; Ahn, M.; Gim, J.; Kim, J.; Jung, J.; Choi, K.; Pham, V.; Rebholz, O.; Malladi, K. T.; Ki, Y.: Optimizing Data Movement with Near-Memory Acceleration of In-memory DBMS. In: Proceedings of the 23rd International Conference on Extending Database Technology. EDBT '20, pp. 371–374, 2020, DOI: 10.5441/002/edbt.2020.35.

[LFS24] Lütke Dreimann, M.; Friesel, B.; Spinczyk, O.: HetSim: A Simulator for Task-based Scheduling on Heterogeneous Hardware. In: Companion of the 15th ACM/SPEC International Conference on Performance Engineering. ICPE '24 Companion, Association for Computing Machinery, pp. 261–268, 2024, DOI: 10.1145/3629527.3652275.

[Li23] Lim, C.; Lee, S.; Choi, J.; Lee, J.; Park, S.; Kim, H.; Lee, J.; Kim, Y.: Design and Analysis of a Processing-in-DIMM Join Algorithm: A Case Study with UPMEM DIMMs. Proc. ACM Manag. Data 1 (2), 2023, DOI: 10.1145/3589258.

[LRJ18] Li, S.; Reddy, D.; Jacob, B.: A performance & power comparison of modern high-speed DRAM architectures. In: Proceedings of the International Symposium on Memory Systems. MEMSYS '18, Association for Computing Machinery, pp. 341–353, 2018, DOI: 10.1145/3240302.3240315.

[Mc04] McKee, S. A.: Reflections on the memory wall. In: Proceedings of the 1st Conference on Computing Frontiers. CF '04, Association for Computing Machinery, p. 162, 2004, DOI: 10.1145/977091.977115.

Tutorial: Programming Processing-in-Memory for Data Management

Kai-Uwe Sattler [1] and Muhammad Attahir Jibril [2]

Abstract: Processing-in-Memory (PIM) is a paradigm promising to reduce data movement as a growing bottleneck in data-intensive systems such as database systems. The main idea of PIM is to bring computation to memory, requiring to rethink architectures and designs of data management solutions. In this tutorial, we give an overview of the application of the PIM paradigm to data management tasks using the UPMEM PIM technology.

1 Motivation

Traditional computer systems are processor-centric: all computations are performed by a CPU (or additional processors such as GPU), which means that other components such as memory, storage, or network are more or less only "suppliers" or "depots" for data. As a result, data are constantly transferred from these components to the processor for computation. With the ever-growing data sizes and analysis requirements of modern applications, together with the increasing capacity of memory, storage, etc., this transfer represents a big factor with respect to performance and cost [Mu20]. Hence, main memory, particularly, becomes more and more a performance bottleneck.

Whereas improving bandwidth, latency, and capacity of DRAM at the same time is very difficult or not even possible, a viable solution to reduce the data transfer is to bring computation to the data. In addition to smart SSDs for storage and smart NICs for networking, the Processing-in-Memory (PIM) paradigm is a promising approach. Although this idea goes back to the 1960s [Ka69], advances in chip technology now enable the realization of the vision to make memory an active unit with computing capabilities in real products.

Database systems are inherently data intensive, and data transfer plays an important role: between external storage and main memory (via a bufferpool), between main memory and CPU caches and registers for most query operators, or even between host and device memory for accelerators. Therefore, reducing transfer costs is a key factor in improving performance, and applying the PIM paradigm is also an interesting approach for database systems. However, exploiting PIM in database systems requires rethinking their architectural aspects, as well as the careful design of data structures and access methods. For this, understanding the features, opportunities, and pitfalls of current PIM technology is necessary.

1 TU Ilmenau, Database & Information Systems Group, P.O. Box 100 535, 98684 Ilmenau, Germany, kus@tu-ilmenau.de, https://orcid.org/0000-0003-1608-7721
2 TU Ilmenau, Database & Information Systems Group, P.O. Box 100 535, 98684 Ilmenau, Germany, muhammad-attahir.jibril@tu-ilmenau.de, https://orcid.org/0000-0003-2138-881X

doi:10.18420/BTW2025-141

In this tutorial, we plan to give an overview of UPMEM PIM, one of the currently commercially available PIM technologies, and introduce the basic steps of programming with PIM. We hope that participants will gain first insights into PIM programming from a database perspective.

2 PIM for Database Systems

The PIM paradigm can be interpreted in a broader range: from rather simple bulk operations (e.g. bitwise operations) on rows of memory cells, requiring minimal changes to existing DRAM – a paradigm that is called Processing-using-Memory in [Mu20] – to adding PIM capabilities to the logic layer of 3D-stacked memory such as in 3D HB-PNM [Ni22] and integrating compute cores with the memory banks in DRAM, e.g. UPMEM PIM [De19].

There are several approaches aiming to adapt database operations to the PIM paradigm. Earlier work was based on hardware simulation or PIM emulation, making it difficult to transfer the findings to real hardware, simply because the assumptions regarding architecture and performance as well as the behavior exhibited by simulators are different from the real products [AS19]. Only a few approaches make use of real PIM hardware such as UPMEM [De19], the first commercially available PIM-enabled hardware. One of the first work was an extensive benchmark [Gó22]. Other approaches consider scan-select operations [BKP23], joins [Li23]. In our own previous work we have investigated data transfer [BJS23b] and adapted database scans [BJS23a] as well as aggregation operators [JAS24].

In addition to UPMEM's PIM technology, a few other solutions have been announced (e.g. AxDIMM [Le22] and Samsung's HBM-PIM [Ki23]), but are not considered in this tutorial.

3 Programming UPMEM PIM

3.1 Hardware Description

An UPMEM DIMM is based on a standard DDR4 DIMM that is compute-enabled via an integrated RISC processor called DPU (*DRAM Processing Unit*). As shown in Fig. 1, each module is organized into 2 ranks, each rank contains 8 PIM chips, and each PIM chip has 8 DPUs. A DPU has access to a 64-MiB *Main RAM* (MRAM), a fast 64-KiB scratchpad called *Working RAM* (WRAM) and a 24-KiB *Instruction RAM* (IRAM). DPUs have access only to their own MRAM, i.e. there is no direct communication possible between different DPUs and all data transfers between DPUs have to go via the CPU. However, both WRAM and MRAM are shared with the host CPU.

A DPU is a 32-bit in-order RISC core executing a special instruction set with up to 24 hardware threads that share the IRAM and the WRAM, and each thread has 24 32-bit

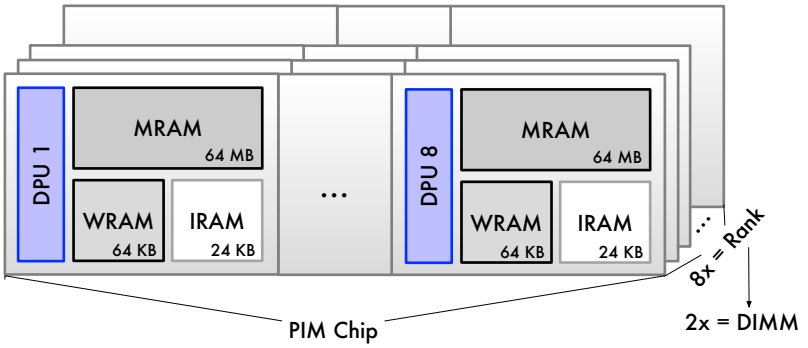


Fig. 1: Architecture of UPMEM DIMM

general-purpose registers. In this way, up to 24 software threads called *tasklets* are supported. Tasklets execute the same program called DPU *kernel* on different data, use synchronization primitives for access control, and execute different control flow paths at runtime. In the current version, DPUs run at a clock speed of up to 400 MHz. While this is slow compared to host CPUs, the biggest advantage of UPMEM's technology is the direct access to MRAM and WRAM from the DPU without the need to involve the CPU, resulting in a high aggregate MRAM bandwidth of up to 1.5 TB/s. Currently, a PIM-equipped system can accommodate 20 PIM DIMMs (alongside conventional DRAM DIMMs) with a total of 160GB of MRAM and 2560 DPUs.

3.2 Programming Model and Tools

The PIM programming model is based on the Single Program Multiple Data (SPMD) paradigm, with fine-grained multithreading among tasklets. A PIM application consists of a host program and a DPU program. The host program coordinates the execution of the DPU program by the DPUs as well as the communication between the CPU and the DPUs. UPMEM provides a set of tools and libraries for building PIM applications. The programming model is based on kernels written in C and compiled by a special compiler, which is based on Clang/LLVM (Listing 1). The host and DPU runtime libraries provide support for allocating the DPUs, data transfer between the CPU and DPUs, MRAM access and management, synchronization of tasklets etc. Other tools include `dpu-lldb`, a debugger, which can additionally be used to execute DPU binaries directly.

```
1    dpu-upmem-dpurte-clang -DNR_TASKLETS=16 -o nodmc nodmc.c
```

List. 1: Compiling DPU Program

3.3 Kernel Setup and Execution

The host library provides host APIs for allocating DPUs from one or more ranks, loading pre-compiled kernels on the allocated DPUs, and (synchronously or asynchronously) invoking the DPUs to execute the kernels. The kernel can be executed with up to 24 tasklets (software threads) in parallel, either running it directly using `dpu-lldb` or using the host program as a driver (Listing 2). The number of tasklets must be defined at compile-time. All DPUs and their tasklets execute the same code but on different parts of the data.

```
#include <dpu.h>

#define NR_DPUS 64
#define DPU_BINARY "./nodmc"

int main() {
    struct dpu_set_t set;

    DPU_ASSERT(dpu_alloc(NR_DPUS, NULL, &set));
    DPU_ASSERT(dpu_load(set, DPU_BINARY, NULL));
    DPU_ASSERT(dpu_launch(set, DPU_SYNCHRONOUS));

    DPU_ASSERT(dpu_free(set));

    return 0;
}
```

List. 2: Controlling DPU Program Execution via a Host Program

3.4 Data Transfer and Memory Management

For accessing the MRAM, host APIs are provided by the host library for different kinds of data transfer (e.g. serial, parallel, broadcast, and scatter-gather) between the CPU and the MRAM. Data transfer can also take place between the CPU and the WRAM (Listing 3). However, the CPU-WRAM communication is slower than the CPU-MRAM communication. Since the WRAM serves as a scratchpad memory, the DPU runtime library supports DMA transfer between the MRAM and the WRAM, but with alignment and size constraints.

```
uint32_t dpu_idx;
uint32_t host_results[NR_DPUS];

struct dpu_set_t dpu;
DPU_FOREACH(set, dpu, dpu_idx) {
    DPU_ASSERT(dpu_prepare_xfer(dpu, &host_results[dpu_idx]));
}
DPU_ASSERT(dpu_push_xfer(set, DPU_XFER_FROM_DPU,
    "dpu_result", 0, sizeof(uint32_t), DPU_XFER_DEFAULT));
```

List. 3: Data Transfer between CPU and DPUs

3.5 Synchronization and Communication

The WRAM reserves memory space for global variables and tasklet stacks. The remainder of the WRAM can be used by tasklets as a heap for dynamic allocations and sharing memory. Because MRAM and WRAM are shared among all tasklets on a DPU, the DPU runtime library provides a list of synchronization primitives such as mutexes, semaphores, barriers, and handshakes (Listing 4).

```
#include <defs.h>
#include <alloc.h>
#include <mram.h>
#include <stdint.h>
#include <mutex.h>
#include <barrier.h>

#define NR_ENTRIES 128

MUTEX_INIT(nodmc_mutex);
BARRIER_INIT(nodmc_barrier, NR_TASKLETS);

__host uint32_t dpu_result;
uint32_t __mram_noinit mram_vec[NR_ENTRIES * NR_TASKLETS];

int main() {
    uint32_t tid = me();
    if (tid == 0) {
        mem_reset();
        dpu_result = 0;
    }
    barrier_wait(&nodmc_barrier);

    uint32_t* wram_vec =
        (uint32_t*) mem_alloc(NR_ENTRIES * sizeof(uint32_t));
    mram_read(&mram_vec[tid * NR_ENTRIES],
        wram_vec, NR_ENTRIES * sizeof(uint32_t));

    uint32_t res = reduce_func(wram_vec);
    mutex_lock(nodmc_mutex);
    dpu_result += res;
    mutex_unlock(nodmc_mutex);

    return 0;
}
```

List. 4: A DPU Program

Overall, the workflow of a host program running a DPU kernel with UPMEM technology can be described in the following steps:

(1) DPU resource allocations (DPUs, ranks, kernel),

(2) buffer population from the host's main memory to MRAM of DPUs,
(3) execution of the DPU kernel, and
(4) retrieval of the processed results from the MRAM of the DPUs to the host's main memory.

Moreover, it is possible to execute multiple iterations of steps (2) – (4) when executing a DPU kernel. The data remain in the MRAM and WRAM of the DPUs and do not have to be reinitialized. This is useful for tasks where a solution has to be calculated in several iterations.

4 Tutorial Outline

The goal of this tutorial is to give an introduction to practical programming with PIM, exemplified with UPMEM technology. Participants will learn the basic steps of PIM programming necessary to understand and apply the paradigm, as well as to be able to decide whether PIM is a viable choice for their own data management research problem. During the tutorial, we will use the UPMEM SDK [UPMEM24] together with a simulation environment, allowing participants to execute their own programs on-site or later at home. The tutorial will cover the following steps:

1. Introduction and overview: Programming model and tools
2. Kernel setup: How to set up the kernel of a UPMEM program
3. Execute: How to start and execute a UPMEM kernel from a host program
4. Data transfer: How to transfer data from host to UPMEM memory and back
5. Data layout and memory access
6. Synchronisation and communication
7. Implementing an aggregation scan

As an example, we will implement a basic database task to scan and aggregate data.

Acknowledgements. This work was partially funded by the German Research Foundation (DFG) in the context of the project "Processing-In-Memory Primitives for Data Management (PIMPMe)" as part of the priority program "Disruptive Memory Technologies" (SPP 2377, SA 782/31).

References

[AS19] Akram, A.; Sawalha, L.: A Survey of Computer Architecture Simulation Techniques and Tools. IEEE Access 7, pp. 78120–78145, 2019, DOI: 10.1109/ACCESS.2019.2917698.

[BJS23a] Baumstark, A.; Jibril, M. A.; Sattler, K.: Accelerating Large Table Scan Using Processing-In-Memory Technology. Datenbank-Spektrum 23 (3), pp. 199–209, 2023, DOI: 10.1007/S13222-023-00456-Z.

[BJS23b] Baumstark, A.; Jibril, M. A.; Sattler, K.: Processing-in-Memory for Databases: Query Processing and Data Transfer. In (May, N.; Tatbul, N., eds.): Proceedings of the 19th International Workshop on Data Management on New Hardware, DaMoN 2023, Seattle, WA, USA, June 18-23, 2023. ACM, pp. 107–111, 2023, DOI: 10.1145/3592980.3595323.

[BKP23] Bernhardt, A.; Koch, A.; Petrov, I.: pimDB: From Main-Memory DBMS to Processing-In-Memory DBMS-Engines on Intelligent Memories. In (May, N.; Tatbul, N., eds.): Proceedings of the 19th International Workshop on Data Management on New Hardware, DaMoN 2023, Seattle, WA, USA, June 18-23, 2023. ACM, pp. 44–52, 2023, DOI: 10.1145/3592980.3595312.

[De19] Devaux, F.: The true Processing In Memory accelerator. In: 2019 IEEE Hot Chips 31 Symposium (HCS), Cupertino, CA, USA, August 18-20, 2019. IEEE, pp. 1–24, 2019, DOI: 10.1109/HOTCHIPS.2019.8875680, URL: https://doi.org/10.1109/HOTCHIPS.2019.8875680.

[Gó22] Gómez-Luna, J.; Hajj, I. E.; Fernandez, I.; Giannoula, C.; Oliveira, G. F.; Mutlu, O.: Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System. IEEE Access 10, pp. 52565–52608, 2022, DOI: 10.1109/ACCESS.2022.3174101.

[JAS24] Jibril, M. A.; Al-Sayeh, H.; Sattler, K.: Accelerating Aggregation Using a Real Processing-in-Memory System. In: 40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024. IEEE, pp. 3920–3932, 2024, DOI: 10.1109/ICDE60146.2024.00300.

[Ka69] Kautz, W. H.: Cellular Logic-in-Memory Arrays. IEEE Trans. Computers 18 (8), pp. 719–727, 1969, DOI: 10.1109/T-C.1969.222754.

[Ki23] Kim, J. H.; Ro, Y.; So, J.; Lee, S.; Kang, S.; Cho, Y.; Kim, H.; Kim, B.; Kim, K.; Park, S.; Kim, J.; Cha, S.; Lee, W.; Jung, J.; Lee, J.; Lee, J.; Song, J.; Lee, S.; Cho, J.; Yu, J.; Sohn, K.: Samsung PIM/PNM for Transformer Based AI : Energy Efficiency on PIM/PNM Cluster. In: 35th IEEE Hot Chips Symposium, HCS 2023, Palo Alto, CA, USA, August 27-29, 2023. IEEE, pp. 1–31, 2023, DOI: 10.1109/HCS59251.2023.10254711, URL: https://doi.org/10.1109/HCS59251.2023.10254711.

[Le22] Lee, D.; So, J.; AHN, M.; Lee, J.-G.; Kim, J.; Cho, J.; Oliver, R.; Thummala, V. C.; JV, R. s.; Upadhya, S. S.; Khan, M. I.; Kim, J. H.: Improving In-Memory Database Operations with Acceleration DIMM (AxDIMM). In: Proceedings of the 18th International Workshop on Data Management on New Hardware. DaMoN '22, Association for Computing Machinery, Philadelphia, PA, USA, 2022, ISBN: 9781450393782, DOI: 10.1145/3533737.3535093.

[Li23] Lim, C.; Lee, S.; Choi, J.; Lee, J.; Park, S.; Kim, H.; Lee, J.; Kim, Y.: Design and Analysis of a Processing-in-DIMM Join Algorithm: A Case Study with UPMEM DIMMs. Proc. ACM Manag. Data 1 (2), 2023, DOI: 10.1145/3589258.

[Mu20] Mutlu, O.; Ghose, S.; Gómez-Luna, J.; Ausavarungnirun, R.: A Modern Primer on Processing in Memory. CoRR abs/2012.03112, 2020, URL: https://arxiv.org/abs/2012.03112.

[Ni22] Niu, D.; Li, S.; Wang, Y.; Han, W.; Zhang, Z.; Guan, Y.; Guan, T.; Sun, F.; Xue, F.; Duan, L.; Fang, Y.; Zheng, H.; Jiang, X.; Wang, S.; Zuo, F.; Wang, Y.; Yu, B.; Ren, Q.; Xie, Y.: 184QPS/W 64Mb/mm^{2}3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System. In: IEEE International Solid-State Circuits Conference, ISSCC 2022, San Francisco, CA, USA, February 20-26, 2022. IEEE, pp. 1–3, 2022, DOI: 10.1109/ISSCC42614.2022.9731694, URL: https://doi.org/10.1109/ISSCC42614.2022.9731694.

[UPMEM24] UPMEM SDK, 2024, URL: https://sdk.upmem.com/.

Lazy DBMS Storage Design with Computational Storage

Alexander Baumstark [1] and Kai-Uwe Sattler [1]

Abstract: The size of datasets in DBMSs is growing super-linearly, increasing the demand for storage devices as data can no longer fit entirely in the main memory. Data movement between the CPU and storage devices often limits performance, primarily due to poor data locality, which is difficult to optimize in DBMSs. An emerging solution to this challenge is the use of Computational Storage Devices (CSDs), which offload certain computation tasks to the storage hardware. However, due to the high cost of fully programmable CSDs, current solutions often offer limited computation capabilities, such as the Scaleflux CSD-3000, which features transparent compression. Transparent compression allows for more flexibility in the storage layout of data structures and DBMS storage formats. As a result, traditional techniques for reducing storage overhead in B^+-Trees can be simplified or eliminated, leading to higher bandwidth and reduced storage overhead. In this work, we evaluate the CSD-3000 with transparent compression by examining commonly used DBMS data structures and storage layouts. Our experiments cover a variety of DBMSs with different storage formats. The results demonstrate that transparent compression reduces storage overhead while improving bandwidth through better data locality.

Keywords: Computational Storage, Data Structures, Transparent Compression

1 Introduction

A key trend in modern system architectures is offloading computation to where the data resides. Technologies facilitating this include processing-in-memory (PIM) [Mu22] with accelerators such as GPUs or FPGAs, interconnects like CXL for efficient resource decoupling and sharing [Ba24], and smart storage devices [Vi20]. These approaches aim to minimize unnecessary data movement within the system, enhancing overall efficiency and performance as more resources are available for the required data. Generally, poor data locality in DBMS is a direct cause of excessive data transfers from and to storage and is hard to accomplish efficiently. For relational DBMS, non-optimal data locality directly affects join processing since access to specific rows is an important driver for join performance. This issue is particularly prevalent in graph DBMSs, where achieving data locality is inherently challenging. Special data structures such as Compressed Sparse Row (CSR) structures are required for optimal locality but come with high update handling costs. For analytical workloads with scan-heavy tasks, a large portion of transferred data is sent to the CPU to be discarded. For transactional workloads, I/O amplification worsens the problem. Data, usually organized into KB-sized pages, contain multiple fixed-sized records. When updating a single record, the entire page must be loaded from storage into memory, updated,

[1] TU Ilmenau, DBIS, Helmholtzplatz 5, 98693 Ilmenau, Germany,
alexander.baumstark@tu-ilmenau.de, https://orcid.org/0000-0002-7162-7615;
kus@tu-ilmenau.de, https://orcid.org/0000-0003-1608-7721

doi:10.18420/BTW2025-142

and then written back to persistent storage. This process mostly wastes I/O time on data that remains unused.

Emerging computational storage devices (CSD) offer a promising solution to mitigate these challenges by minimizing unnecessary data movement and improving efficiency through near-data processing [Hu23]. These devices integrate processing units with dedicated memory, enabling computational tasks to be performed directly on the storage medium alongside standard storage operations. From an economic perspective, equipping drives with more computation capabilities, such as fully programmable FPGAs, to offload certain more complex computation tasks to storage, increases the cost of CSDs. Additionally, these technologies introduce higher complexity to database management systems (DBMS). The hardware design of modern SSDs relies on a robust processing unit with high resource utilization in both computation and memory. However, integrating FPGAs often results in underutilized memory and storage resources, increased energy consumption, higher wear leveling, and reduced device lifespan. Consequently, as of this writing, there are no commercially available CSDs with fully programmable capabilities beyond prototype FPGA-based solutions [Xi24]. Technologies like CXL and other PCIe-based solutions (e.g., GPUDirect, NVLink) present opportunities to address these challenges efficiently.

In practice, CSDs are more cost-effective when equipped with non-programmable processing units, or a System on a Chip (Soc), such as the ScaleFlux CSD [Sc24]. This device supports transparent compression, a promising approach to mitigate I/O amplification and improve data locality. Based on the underlying hardware design of the Scaleflux CSD with transparent compression, we identified the following opportunities for DBMS which consequently helps to unwind the underlying storage design of a DBMS.

1. Log/text compression: Logs or text are often already compressed via standard compression techniques and carried by the CPU. These structures comprise repetitive symbols and are perfectly suitable for compression.
2. Sparse data structure compression: Indexes in DBMSs are often sparsely designed or occupied, either intentionally to postpone costly operations (B+-Tree), or due to low data occupation (Hash map). Transparent compression enables storing such structures more efficiently since unused (zeroed) data is effectively compressible.
3. Row-based/ key-value layout compression: Row-based layout and key-value layouts are hard to compress with techniques such as RLE or delta encoding. CSDs with transparent compression offer a promising solution to overcome this problem.

For storage with transparent compression, we expect a reduced physical storage space occupation for sparse data structures and a higher bandwidth than on usual storage devices.

2 Transparent Compression with Computational Storage

2.1 Properties

The Scaleflux CSD-3320 integrates a hardware engine to (de)compress a 4 KB data block. The Flash Translation Layer manages the storage of the variable-sized compressed data blocks in NAND flash memory. The compression of 4 KB data blocks is processed without CPU intervention, reducing CPU load effectively while reducing storage size transparently. Compared to similar SSDs without transparent compression, the CSD 3000 comes with a much higher IOPS with 2:1 compressible data [Hu23]. This shows the significance of optimized data layout for compression to achieve maximum performance of the SSD. In usual NAND-based memory SSD, the logical page number is assigned directly to a physical page number so that a 4 KB data block is written directly to a physical page on NAND memory. With transparent compression, 4 KB pages are compressed before being written to NAND memory. Therefore, multiple logical data blocks can be written to a single physical NAND memory page.

2.2 In-device Transparent Compression

The Scaleflux CSD-3320 device employs a hardware engine featuring a SoC with an integrated compression mechanism based on the zlib compression scheme, which uses the DEFLATE algorithm [Ha14]. The DEFLATE algorithm combines two primary techniques: LZ77 for finding and encoding repeated patterns and Huffman Coding for reducing the size of the encoded data by using shorter codes for more frequently occurring patterns. The internal SoC of the Scaleflux CSD-3000 processes data in 4 KB blocks, compressing them into smaller, more space-efficient blocks. During reads, these compressed blocks are identified and decompressed back into the original 4 KB size. This process is transparent to the host, meaning the host system interacts with the data as if it were uncompressed. The algorithm operates best when data is repetitive, as the LZ77 algorithm replaces these substrings with references to earlier occurrences, on long runs of identical data (similarly to RLE), text, and structure data due to the recurring patterns.

2.3 Data Structures

B$^+$-Tree The B$^+$-Tree is the most widely adopted index structure in DBMSs. It primarily consists of two components: a persistent store for the leaf nodes and an in-memory store for the inner nodes. Leaf nodes, which contain the actual data, are typically allocated in fixed-size pages. Inner nodes serve as an index to the leaf nodes, enabling efficient traversal during read and write operations. Every operation begins by navigating through the inner nodes to locate the appropriate leaf node. Insert operations are performed directly on the leaf

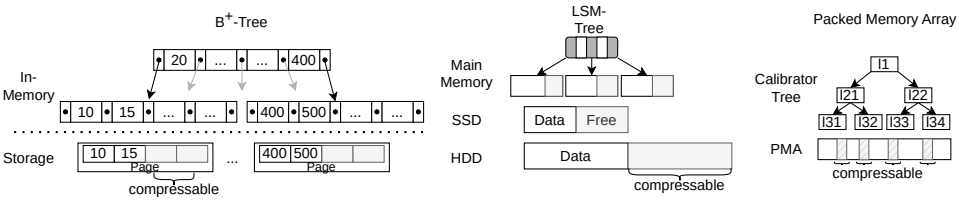


Fig. 1: Common data structures and their internal representation in storage.

nodes if there is sufficient space. If the leaf node is full, a node split operation is triggered. This splits the node into two, redistributes the data evenly between them, and inserts a new index entry into the parent node. This process may propagate recursively up the tree until all nodes are within their size limits [Hu23]. When nodes become underutilized (i.e., fall below a predefined threshold), they are merged with neighboring nodes to optimize space. However, this leads to a common issue: after a split, both resulting nodes typically have a fill factor of only 50%, causing significant storage inefficiency, as shown in Fig. 1. Although various solutions exist to mitigate these issues, they often introduce additional complexity and computational overhead. The opportunity with transparent compression with CSD lies in the effective storage of underutilized leaf nodes which reduces the storage overhead. Further, complex tree structure optimization is not necessary anymore, reducing the computation overhead.

LSM-Tree The Log-Structured Merge Tree (LSM-Tree) has gained popularity in storage systems due to its performance and efficient storage utilization compared to B$^+$-Trees [ON96]. An LSM-Tree consists of multiple levels, denoted as $L_0, \ldots, L_n$. The L_0 level resides in the main memory, while subsequent levels are stored in different storage tiers, such as disk or flash memory. The L_0 level is typically implemented as an in-memory B$^+$-Tree or a Skip List and serves as a buffer for batching write operations. When the data in L_0 reaches a certain threshold, it is flushed to L_1. This process, known as compaction, is recursively applied to subsequent levels, merging and reorganizing data as it moves further down the hierarchy. While LSM-Trees achieve higher density compared to B$^+$-Trees, they require doubling the storage allocation at each level, which can lead to additional storage overhead. Similar to the B$^+$-Tree, transparent compression allows storing empty space on the upper levels more effectively. This reduces the overall costs for pre-allocating larger space.

Packed Memory Array Packed Memory Arrays (PMA) maintain data in sorted order while enabling efficient insertions by strategically preserving gaps within the data structure. These gaps reduce the need for extensive reordering when new elements are inserted between existing ones. PMA supports searching through a binary search, which effectively skips over the gaps. A PMA can be used for efficient graph storage, based on a CSR. When inserting

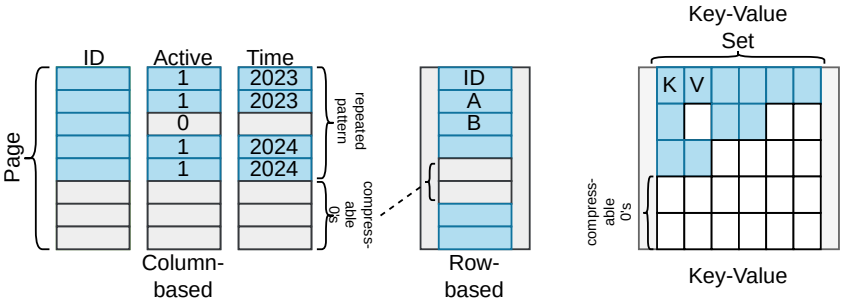


Fig. 2: Different storage layout formats for DBMS and the opportunities for transparent compression.

an element, the appropriate position is located via binary search. If the identified slot is empty, the element is inserted directly. If the slot is occupied, the element is shifted towards the nearest available gap. Deletions in a PMA are typically processed logically, marking elements for removal rather than immediately reorganizing the structure. Optimizing PMA for storage efficiency requires sequentially writing data and re-inserting it when read, which increases computational overhead. Alternatively, storing data directly without reorganization increases storage overhead due to inefficient gap utilization. When storing such structures directly, transparent compression can reduce the inherent storage overhead of the gaps, making PMA more efficient.

Logs A DBMS ensures a consistent state even after failures of any kind. Logging is the most widely employed technique to achieve this consistency. Logging records all executed operations, along with metadata such as timestamps, transaction IDs, previous and next entries, and information necessary to undo or redo operations. This allows the system to recover to a consistent state in the event of a failure. Log files typically exhibit recurring patterns. To mitigate the growth of log file size, which increases with the number of operations, many DBMSs employ standard compression techniques, such as prefix compression.

2.4 Data Layouts

DBMS are designed to meet specific workload requirements. OLTP workloads benefit from a row-based layout because such transactions often comprise reading or writing complete records instead of accessing single attributes. OLAP workloads typically favor a column-based layout, enabling efficient processing of analytical queries that require scanning large datasets often over single attributes, i.e., for aggregations. Key-value workloads excel with a key-value store layout, which optimizes rapid lookups, inserts, and updates by

efficiently retrieving values based on unique keys. The storage layout of the DBMS affects the achievable compression rate of a CSD with transparent compression.

Column-based In a column-based format (Fig. 2 left), the data for each column (attribute) are stored separately in fixed-size pages. As a result, attributes of the same type and size are stored sequentially across multiple pages within a single file. This approach offers the advantage of enabling efficient compression techniques. RLE can be effectively applied when similar data appear consecutively. Delta Encoding is beneficial when there are small differences between consecutive entries. These compression methods reduce storage requirements and improve query performance by minimizing the amount of data read.

Row-based In a row-based storage format in Fig. 2 (middle), the data for each row (a record) is stored together sequentially within fixed-size pages. As a result, all attributes of a row, regardless of type and size, are stored contiguously, facilitating quick access to the entire record. This approach is advantageous for workloads that frequently access or update entire rows, such as transactional processing (OLTP) systems. Since all fields of a row are stored together, queries that require retrieving all attributes of a record can be processed efficiently with minimal disk I/O. However, row-based storage is generally less efficient for compression compared to column-based storage. Attributes of varying types and sizes are interleaved, making it difficult to apply techniques like Run-Length Encoding (RLE) or Delta Encoding. Consequently, compression ratios tend to be lower, and the system may need to read more data than necessary for queries that access only a subset of the attributes. While row-based storage can be suboptimal for analytical queries (OLAP), it excels in scenarios where entire rows are frequently accessed or modified.

Key-value In a key-value store, such as in Fig. 2 (right), data is organized as a collection of key-value pairs, where each unique key is associated with a corresponding value. These pairs are stored sequentially within fixed-size pages or blocks, depending on the underlying storage engine. This storage approach offers significant flexibility, as values can be of arbitrary types and sizes, such as strings, JSON objects, or binary data. Accessing data in key-value stores is highly efficient for workloads that involve frequent lookups, updates, or deletions based on keys, making them ideal for high-throughput transactional workloads. Key-value stores often support simple and effective compression techniques. When keys exhibit similar prefixes, techniques like Prefix Compression can be applied to reduce storage requirements. For values, methods such as zlib compression can efficiently reduce the data footprint, especially if the values contain redundant or repetitive data. Since key-value stores do not impose a fixed schema, optimizing for analytical queries or complex operations involving multiple keys may be less efficient than optimizing for structured databases. The layout limits the applicability of compression methods like RLE or Delta Encoding. However, the use of similar keys can employ dictionary encoding-based compression effectively.

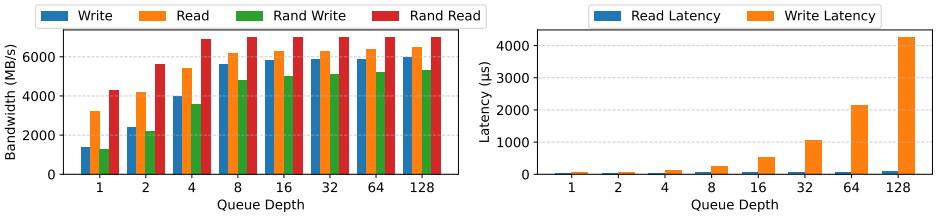


Fig. 3: Sequential bandwidth and total access latency of the Scaleflux CSD-3320.

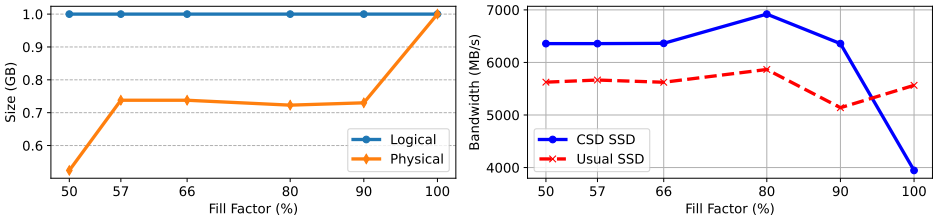


Fig. 4: Compressed B$^+$-Tree sizes and sequential bandwidth on a CSD with transparent compression.

3 Experiments

For the following benchmarks, we use the Scaleflux-3320 NVMe SSD, connected via PCIe 4.0. It provides 3.84 TB capacity, which can be logically extended up to 9 TB using thin provisioning. We formatted the Scaleflux drive with an ext4 filesystem. For comparison, we use a usual NVMe SSD (Kioxia XG8) with 1 TB, also connected via PCIe 4.0. The host system has two Intel Xeon Max 9462 and 256 GB of DDR5 main memory and runs on Debian 12.8 on Linux 6.10. To obtain the internal compression sizes and rates, we use the tools provided by Scaleflux, i.e., `sfx-filesize`.

Device Fig. 3 left shows bandwidth and latency numbers of the CSD-3320 device with transparent compression. The device is connected via PCIe 4.0 and achieves a bandwidth of up to 7 GB/s when reading and around 4GB/s when writing data, which lies in the same range as a comparable NVMe SSD without transparent compression such as the Kioxia XG8 used in our system. Fig. 3 right shows the access latency with increasing queue depth size, which is the number of pending requests to the device controller. The latency remains the same for read operations while increasing with the number of writes. The CSD-3000 compression engine has a 5μs latency for both compression and decompression [Hu23], making it an efficient storage solution for a DBMS.

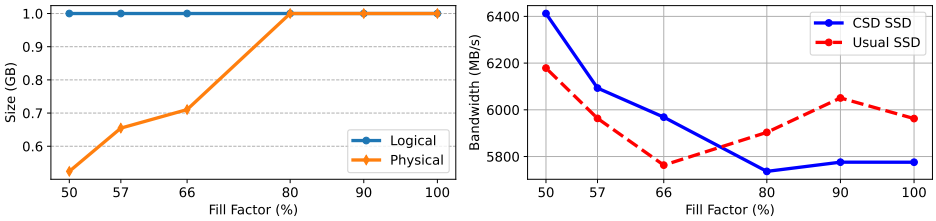


Fig. 5: Compressed LSM-Tree sizes and sequential bandwidth on a CSD with transparent compression.

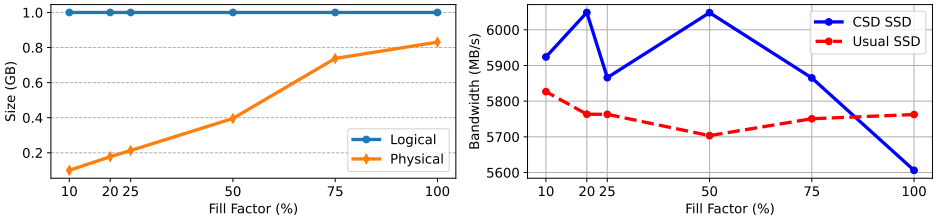


Fig. 6: Compressed PMA sizes and sequential bandwidth on a CSD with transparent compression.

Data Structures We selected a B$^+$-Tree with a 4096 KB leaf node size and populated it with random data. Fig. 4 provides a detailed view of the resulting compressed storage footprint (left) and the read bandwidth of compressed data (right). The random data used has a similarity rate of approximately 5% across all fill factors; higher similarity would yield better compression rates. As the fill factor increases, the compression rate decreases due to the limitations of the DEFLATE algorithm in compressing less redundant data. However, since B$^+$-Trees are typically maintained with a low fill factor, fill factors in the 50% to 80% range reflect realistic use cases. The bandwidth comparison shows the read performance of a CSD compared to a standard NVMe SSD without transparent compression. When the compression rate is high, the CSD achieves higher bandwidth due to improved data locality. Compressed pages are stored sequentially, enhancing sequential read performance. Conversely, as the compression rate decreases, the sequential bandwidth decreases below that of an SSD without transparent compression.

For the LSM-Tree, we use an SSTable with two levels, where each level allocates the doubled size of the previous level. Again, we filled it out with random data. Fig. 5 shows the fill factor (per level) for an LSM-Tree and its sequential read bandwidth. Here, an LSM-Tree with a total storage occupation of 1 GB can be compressed to 480 MB. However, the LSM-Tree is stored more densely on storage. Therefore, compression can only reduce the allocated but unused space. The difference in compression rate, compared to B$^+$-Tree is due to the position of not filled space in the layout. Since the compression rate is low, bandwidth is lower than when using a sparse B$^+$-Tree.

For the PMA, we store random data in a 1 GB allocated PMA. Compressed sizes for different fill factors and sequential bandwidths are shown in Fig. 6. Since the PMA is a

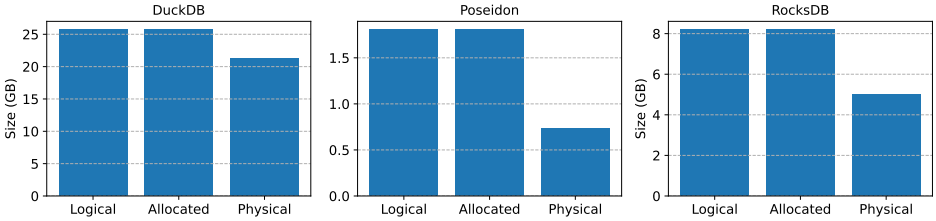


Fig. 7: Storage space occupation for DuckDB and Poseidon without transparent compression (logical) which is the actual allocated size, and its actual size with transparent compression (physical).

sparse data structure with gaps between entries, it can be compressed efficiently. Similar to the LSM-Tree, the difference in compression rate, compared to B^+-Tree is due to the position of not filled space in the layout. Again, with a decreasing compression rate, the sequential bandwidth decreases.

DBMS Storage For the evaluation of different DBMS storage formats, we use three systems: DuckDB[2], which employs a column-based layout; Poseidon[3], a graph DBMS with a row-based storage format; and RocksDB[4], representing the key-value store category. The resulting storage sizes are shown in Fig. 7. For DuckDB, we use the TPC-H dataset with a scale factor of 100, which occupies 26 GB of storage without compression. By enabling transparent compression, the physical storage requirement is reduced to 22 GB, achieving a compression ratio of 1.21. This compression efficiency is primarily due to the prevalence of similar data patterns in the dataset. As data size increases, the likelihood of encountering similar patterns rises, enhancing compression efficiency. Additionally, DuckDB's columnar layout facilitates effective compression methods, such as RLE for integer columns.
For Poseidon, we utilize the GTPC dataset [JBS23], which is derived from TPC-H. We use a scale factor equivalent to 50 warehouses, resulting in a storage size of 1.7 GB without compression. Poseidon applies dictionary compression to handle string data more efficiently. With transparent compression applied to the CSD, the physical storage requirement is reduced to 0.75 GB. Although the row-based layout typically hinders compression (e.g., RLE is difficult to apply effectively), transparent compression enables efficient data compression even in row-based format.
For RocksDB, we insert 8 GB of randomly generated key-value pairs. When this data is placed on a CSD with transparent compression, the physical storage size is reduced to 4.5 GB. This reduction is due to RocksDB's underlying storage layout, which uses SSTables and maintains data in sorted order. Consequently, similar data is clustered together, allowing compression algorithms to achieve higher efficiency.

[2] https://github.com/duckdb/duckdb
[3] https://github.com/dbis-ilm/poseidon_core
[4] https://github.com/facebook/rocksdb

4 Conclusion

Computational storage is now readily available for DBMS use, eliminating the need for expensive operator offloading to storage. Transparent compression allows DBMSs to adopt less efficient storage layouts because certain types of data, such as text or sparse data, can be effectively compressed. This is particularly beneficial for B^+-Trees with low fill factors or other sparse data structures. Transparent compression improves storage utilization and enhances bandwidth by improving data locality. While similar benefits can be achieved with denser structures, they often require additional effort and depend heavily on the nature of the stored data. In general, structured data tends to be more compressible. Our results demonstrate that row-based storage layouts can be as suitable for compression as column-based layouts, contrary to conventional assumptions. Future work should explore additional data structures and optimizations in storage representation to further improve compression rates and overall storage efficiency. Further, the Flexible Data Placement (FDP) NVMe feature should be explored with the upcoming Scaleflux CSD-5000 devices.

Bibliography

[Ba24] Baumstark, Alexander; Paradies, Marcus; Sattler, Kai-Uwe; Kläbe, Steffen; Baumann, Stephan: So Far and yet so Near-Accelerating Distributed Joins with CXL. In: Proceedings of the 20th International Workshop on Data Management on New Hardware. pp. 1–9, 2024.

[Ha14] Harnik, Danny; Khaitzin, Ety; Sotnikov, Dmitry; Taharlev, Shai: A fast implementation of deflate. In: 2014 Data Compression Conference. IEEE, pp. 223–232, 2014.

[Hu23] Huang, Kecheng; Shen, Zhaoyan; Shao, Zili; Zhang, Tong; Chen, Feng: Breathing New Life into an Old Tree: Resolving Logging Dilemma of B+-tree on Modern Computational Storage Drives. Proceedings of the VLDB Endowment, 17(2):134–147, 2023.

[JBS23] Jibril, Muhammad Attahir; Baumstark, Alexander; Sattler, Kai-Uwe: GTPC: Towards a Hybrid OLTP-OLAP Graph Benchmark. In: BTW'23. pp. 105–117, 2023.

[Mu22] Mutlu, Onur; Ghose, Saugata; Gómez-Luna, Juan; Ausavarungnirun, Rachata: A modern primer on processing in memory. In: Emerging computing: from devices to systems: looking beyond Moore and Von Neumann, pp. 171–243. Springer, 2022.

[ON96] ONeil, Patrick; Cheng, Edward; Gawlick, Dieter; ONeil, Elizabeth: The log-structured merge-tree (LSM-tree). Acta Informatica, 33:351–385, 1996.

[Sc24] Scaleflux: Scaleflux CSD-3000. https://scaleflux.com/products/csd-3000/, 2024. Accessed: 2024-12-15.

[Vi20] Vincon, Tobias; Bernhardt, Arthur; Petrov, Ilia; Weber, Lukas; Koch, Andreas: nKV: near-data processing with KV-stores on native computational storage. In: Proceedings of the 16th International Workshop on Data Management on New Hardware. pp. 1–11, 2020.

[Xi24] Xie, Rui; Ma, Linsen; Zhong, Alex; Chen, Feng; Zhang, Tong: ZipCache: A DRAM/SSD Cache with Built-in Transparent Compression. arXiv preprint arXiv:2411.03174, 2024.

Dreaming of Syscall-less I/O with io_uring

Some Assembly Required, Feaver Dreams and Nightmares included

Constantin Pestka[1] und Marcus Paradies[2]

Abstract:

Over the past decade, storage technology has evolved significantly. Traditional rotational magnetic media, which offered hundreds of IOPS and latencies measured in milliseconds, have been largely replaced by SSDs. Modern SSDs deliver over 10 million IOPS with latencies in the low double-digit microsecond range. This shift has disrupted the assumptions underlying the design of traditional I/O stacks, making them less suitable for modern hardware. To address these challenges, new APIs like NVMe (at the protocol level) and io_uring (at the operating system level) have been developed. These APIs aim to eliminate performance bottlenecks that have become apparent as the hardware has advanced. At the OS level, a key focus of these APIs is reducing the number of system calls, as excessive syscalls can severely impact performance on modern systems by disrupting processor caches. Beyond this, many modern APIs share common design principles, whether they operate at the protocol, OS, or user-space library level. They are typically asynchronous, completion-based, and optimized for high-concurrency I/O operations to fully leverage the capabilities of current hardware. In our proposed tutorial, we will begin by introducing io_uring and explaining its basic functionality. We will highlight its advantages compared to older I/O APIs and conclude with a discussion of the challenges faced by user-space applications seeking to improve performance with io_uring. These include the increased complexity of implementation and scaling issues related to parallelism.

1 io_uring – The Basics

The tutorial will begin with an introduction to io_urings core components such as the Submission Queue (SQ) and Completion Queue (CQ) and give a high level, conceptual description of how io_uring works that will be expanded on in the following sections. We will then explain its basic operational usage from a user perspective based on illustrative examples. This will include a basic introduction to the fundamental challenges and responsibilities of io_uring users. These include its asynchronous nature, the responsibility of the user to match the CQEs of completed requests to initially submitted requests, due to out-of-order arrival of completions and the ordering guarantees, or lack thereof, of the execution of requests.

[2] LMU, Database Systems Group, Oettingenstraße 67, 80538 Munich, Germany, m.paradies@lmu.de
[1] German Aerospace Center, Institute of Data Science, Mälzerstrasse 3, 07745 Jena, Germany, constantin.pestka@c-pestka.de

doi:10.18420/BTW2025-143

2 io_uring – The Good

In this section, we will give an overview of io_urings advantages over traditional I/O APIs, such as regular, non-blocking and/or vectored POSIX I/O, AIO, and epoll. To put the magnitude of these advantages into perspective, we will include performance numbers from various sources, such as [Ax19; Di22; HL23; Le18; RT23], as well as our benchmarking results. The main focus here will be on the comparative reduction or complete elision of syscalls during operation. We will also discuss the true asynchronous nature of io_uring, i.e., guaranteeing to **never** block in any of its related functions or syscalls the user may execute. This is in contrast to AIO, which has much softer guarantees, which is detrimental for many applications, as it interferes with proper user space scheduling. Similarly, we will defer the explanation of how this is achieved in io_uring to the next section. We will end this chapter with an overview of the advantages of io_uring. An incomplete list of the relevant features, which we will briefly describe, respectively, is:

1. Registered file descriptors
2. Registered buffers
3. Multi-shot variants of mostly read-style requests, such as recv, read, accept
4. Zero copy optimizations
5. Bundled operations (for now only send)
6. The ability to cancel in-flight operations
7. Linked requests
8. io_uring exclusive operations e.g. NVMe passthrough, range based fsync

3 io_uring – Internals

As a foundation for the evaluation of some important usage patterns and configurations of io_uring in the following, this section will start to introduce some of the internals of io_uring. Examples of misuse of parameters controlling these configurations are plentiful[2] and often result in high performance degradations. While not the only one, the most important one of these is the configuration of io_uring for fully syscall-less operation, but other parameters, which e.g., allow for the reduction of required *Inter Processor Interrupts* (IPIs), are also notable.

To explain how these configurations of io_uring work, we begin by explaining what tasks have to be performed within io_uring in general. We will then continue by explaining who does what part of said work at which point in time and explain how that changes based

[2] See the discussions on the mailing list or liburings github repository for numerous examples of users not correctly using and understanding these parameters.

on the used configuration. This will most notably include an explanation of how io_uring can achieve fully syscall-less operation via SQ polling and how io_uring guarantees to never block under any circumstances. We will point out which mechanism is used in which situation to achieve this guarantee to never block, as well as their respective performance impacts. This is likely to be of particular interest to the audience, as this has been frequently misunderstood. Many have assumed that io_uring exclusively utilizes the IO WorkQueue, which, ironically, has the highest overhead among the used mechanisms and is hence only used as a last resort. Furthermore, we will include an explanation of how io_uring incorporates I/O device polling and explain both the operations of polled and IRQ driven IO in io_uring.

4 io_uring – The Bad

Based on the details provided in the previous sections, this section we will contrast and discuss the previously listed advantages of io_uring with various trade-offs that have to be made when using io_uring. The first of these trade-offs to be discussed will be, io_urings massively increased complexity compared to any other I/O API. Io_uring's philosophy of exposing many more kernel-internal tuning knobs to allow for much better performance allows for the optimization opportunities that many high-performance application developers, such as DB developers, have demanded previously. While this is indeed a definitive advantage for such developers, correct usage of these features requires intimate understanding of the inner workings of io_uring, but failure to do so will instead result in massive performance regressions or even spurious functional failures, such as deadlocks. This should thus not be underestimated, both in respect to generally increased complexity, but also due to the fact that the documentation for many of these mechanisms and systems is either rather incomplete or even fully absent. To put this point into perspective, we will give some illustrative examples here that can be understood based on the previously explained io_uring internals (e.g., related to `IORING_SETUP_COOPTASKRUN` or `IOSQE_ASYNC`).

We will also include a discussion of some additional potential issues for applications, such as the fundamentally asynchronous nature and thus the need for a high quality in application scheduler and extremely limited support in higher-level libraries, such as libc, seastar, libuv or tokio. The asynchronous nature specifically either tends to be a complete non-issue for applications or requires major or even full rewrites.

5 io_uring – The Ugly

In this section, we will discuss one final difficulty faced by applications when using io_uring, which is of particular interest to database systems. This difficulty comes in the form of the large family of issues faced in the integration of io_uring in highly parallelized systems, such as storage engines of database systems, general purpose I/O libraries, web servers,

and networking backends of distributed database systems. This discussion will be mostly centered around io_urings synchronization methods and their design goals, how these integrate well with some software architectures and notably not so much with others, as well as how certain important features, such as SQ polling are not trivially usable in these scenarios and why that is the case. We will then discuss these more general issues in the context of specific database architectures and touch on some existing integrations, as well as some ongoing efforts and their respective design trade-offs.

6 io_uring – The Best We Have Nonetheless?

The final section will conclude with weighting the previously discussed advantages and disadvantages of io_uring and compare its utility for application domains such as data bases with some of the available such as the traditional APIs and SPDK, as well as some potential future alternatives such as unikernel approaches.

References

[Ax19] Axboe, J.: Efficient IO with io_uring, [Online; accessed 20-October-2023], 2019, URL: https://kernel.dk/io_uring.pdf.

[Di22] Didona, D.; Pfefferle, J.; Ioannou, N.; Metzler, B.; Trivedi, A.: Understanding modern storage APIs: a systematic study of libaio, SPDK, and io_uring. In: Proceedings of the 15th ACM International Conference on Systems and Storage. SYSTOR '22, New York, NY, USA, pp. 120–127, 2022, ISBN: 978-1-4503-9380-5, DOI: 10.1145/3534056.3534945, URL: https://doi.org/10.1145/3534056.3534945, visited on: 06/27/2023.

[HL23] Haas, G.; Leis, V.: What Modern NVMe Storage Can Do, and How to Exploit It: High-Performance I/O for High-Performance Storage Engines. Proc. VLDB Endow. 16 (9), pp. 2090–2102, 2023, ISSN: 2150-8097, DOI: 10.14778/3598581.3598584, URL: https://doi.org/10.14778/3598581.3598584.

[Le18] Leis, V.; Haubenschild, M.; Kemper, A.; Neumann, T.: LeanStore: In-Memory Data Management beyond Main Memory. In: 34th IEEE International Conference on Data Engineering, ICDE. Pp. 185–196, 2018, DOI: 10.1109/ICDE.2018.00026, URL: https://doi.org/10.1109/ICDE.2018.00026.

[RT23] Ren, Z.; Trivedi, A.: Performance Characterization of Modern Storage Stacks: POSIX I/O, Libaio, SPDK, and Io_uring. In: Proceedings of the 3rd Workshop on Challenges and Opportunities of Efficient and Performant Storage Systems. CHEOPS '23, Rome, Italy, pp. 35–45, 2023, ISBN: 9798400700811, DOI: 10.1145/3578353.3589545, URL: https://doi.org/10.1145/3578353.3589545.

Early Career Researcher Workshop

BTW 2025 Early Career Researcher Workshop

Message from the Chairs

Debasree Das[1], Isolde Adler[2], and Daniela Nicklas[1]

We are pleased to organize the BTW 2025 Early Career Researcher Workshop (ECR) as a unique platform dedicated to fostering the next generation of database and information systems researchers. This workshop augments the main conference of BTW 2025 by specifically targeting students, young professionals, and early PhD candidates.

The workshop provides an opportunity to discuss career options in research, training on presentation of research ideas, and have first experience in participating in an academic conference within the database community at the most important database conference in the German-speaking area. This workshop further provides valuable room for gaining first scientific conference experience by presenting research ideas, receiving feedback, and networking with peers. This program helps the participants to have a higher degree of interaction with experts in databases and information systems and also gives an opportunity to find potential PhD supervisors, explore different career paths in BTW-related fields and get research guidance as an early PhD candidate.

Target Group and Selection We selected 25 applicants out of 35 strong applications using our selection criteria of best fit to the target group, current degree programme, research interests and diversity. Out of the 25 participants, we focus on three groups, (i) Graduate Students: MSc students who may consider a PhD in databases or computer science related topics will be able to meet potential supervisors, and explore research opportunities, (ii) Early PhD Candidates: Students who are in their early stage of PhD will be able to present early-stage research for feedback and gain insights to advance their PhD, (iii) Working Professionals: Participants who are already working in industry and considering moving back to academia will be be able to discuss their career options and might find research topics along with potential supervisors.

Program Details On the workshop day, 3rd March, 2025, the program starts by giving a brief introduction about different pathways of pursuing a PhD, continued by an elevator

[1] Chair of Mobile Systems, University of Bamberg, Bamberg, Germany, debasree.das@uni-bamberg.de; daniela.nicklas@uni-bamberg.de
[2] Chair for Algorithms and Complexity Theory, University of Bamberg, Bamberg, Germany, isolde.adler@uni-bamberg.de

doi:10.18420/BTW2025-144

pitch training session on "How to pitch your research idea" by Dr. Claudia Lopez Camara. Following this, we moderate a panel with a set of panelists from different career stages (early to senior) and diverse background of both academia and industry with a focus on the theme "how to get most out of a conference & career opportunities". In order to achieve a high degree of interaction and collaboration among the participants, they prepare posters on their learning throughout the training sessions and panel discussion. On top of the ECR workshop, we provide participation in BTW workshops (March 4th, Tuesday) and the BTW conference (March 5th-7th ; Wednesday-Friday), including the conference dinner. To note down achievements, we also present an achievement card which the participants will be able to showcase during the conference dinner (6th March, 2025). The ECR workshop concludes with a shared lunch following the closing keynote, serving as a retrospective session where participants can reflect on their first conference experience and celebrate their achievements.

Acknowledgment We thank the Gesellschaft für Informatik e.V. (GI), in particular the Fachbereich Datenbanken und Informationssysteme and the Fachgruppe Datenbanken, for providing generous financial support for conducting the program and supporting student travel grants. Another thank goes to the highly motivated applicants for showing their eager enthusiasm and making the program a great success. We are looking forward to meeting them all at BTW 2025 and hope this workshop serves as a valuable stepping stone in their scientific careers, opening new opportunities and insights for their future journeys.

Debasree Das, Isolde Adler, and Daniela Nicklas

Data Science Challenge

Data Science Challenge: Electricity Price Forecasting

This Year With Integrated University Course!

Marco Grawunder [1] and Wolfram Wingerath [1]

Abstract: Just like every year since 2017, this year's BTW 2025 also had a Data Science Challenge to inspire participation from students and practical researchers. Participants dealt with the question of how to predict the hourly prices on the electricity exchange for the coming day. In the fifth edition, we decided to take a fresh approach by integrating the Data Science Challenge into a course that could be conducted by independent instructors with relative ease and that provided participants with a supervision framework and feedback for their submissions. With this approach, we received an increased number of submissions compared to earlier years: 13 teams handed in their results, 5 of which presented them at the conference venue and received a cash prize.

Keywords: Data Science Challenge, BTW

1 New Approach: Challenge With Integrated University Course!

This year, we provided material for professors, research assistants, and other teaching staff to conduct a seminar on the challenge topic, allowing students to develop their submissions during the semester as part of a university course. Our goal was to enhance visibility within the community and increase the number of submissions by making participation more attractive for teachers and students alike.

2 Challenge Goal: Electricity Price Forecasting

Day-ahead prices are the prices set for electricity delivery on the next day, determined through a daily auction process based on anticipated demand and supply. These prices provide a forecast that helps producers and consumers plan their generation and consumption, reflecting expected market conditions. Predicting these prices is crucial for energy companies and traders to optimize their operations, manage risks, and improve market efficiency. As the goal of the BTW 2025 Data Science Challenge, the groups had to build a model to predict hourly day-ahead energy prices for Germany on February 18, 2025 as reported by the SMARD.de information platform (see Fig. 1 for an example showing data for a different date in the past).

[1] University of Oldenburg, Data Science Group, Escherweg 2, 26121 Oldenburg,
marco.grawunder@uol.de, https://orcid.org/0009-0007-7546-0289;
wolfram.wingerath@uol.de, https://orcid.org/0000-0003-3512-5789

doi:10.18420/BTW2025-145

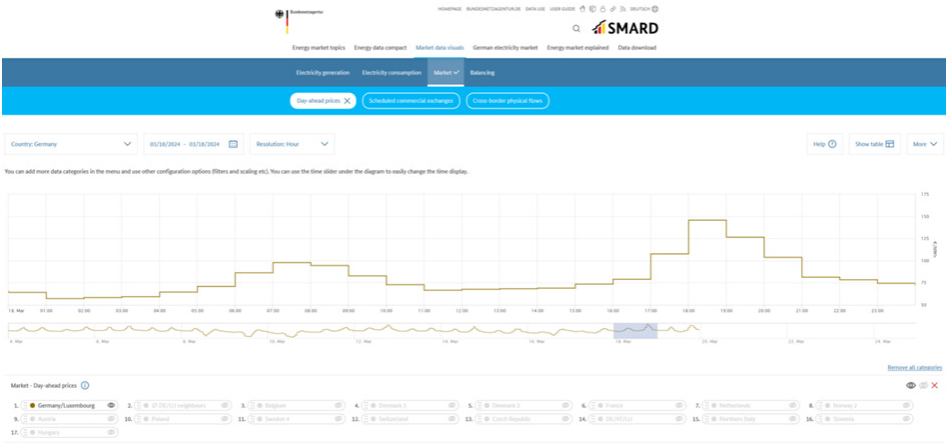


Fig. 1: Day-Ahead Prices for March 18, 2024 as displayed on the SMARD.de information platform.

3 Phases of the Project

The work on the data science challenge was roughly partitioned into the following phases:

- **Gathering Domain Knowledge & Data Sources**: Collect and analyze relevant data sources, build an understanding of the domain.
- **Data Cleaning & Exploratory Data Analysis (EDA)**: Prepare data for a downstream analysis, for example by removing anomalies, investigating variable relationships, and creating statistical summaries.
- **Visualization & Storytelling**: Plot trends and forecasts, evaluate model performance visually, and explain the relevant findings in a compelling narrative.
- **Predictive Modeling**: Develop a forecasting model, compare it against a reasonable baseline, and iterate on it.

4 Suggested Semester Structure

The challenge was structured around a seminar phase, during which participants could attend presentations and engage in discussions to deepen their understanding. This phase was crucial for developing a well-rounded submission. Each team documented their progress and insights in a Jupyter Notebook, divided into the thematic areas specified above. The suggested semester structure is outlined below:

- **Kick-Off (Week 1)**: Presentation by course instructors (with a given slide template), group assignment, selection of seminar topics.

- **Seminar Phase (Weeks 4-6)**: Student presentations on topics relevant for working on the challenge such as:
  - Electricity market pricing mechanisms (Merit-Order, etc.)
  - Power market structures and operators
  - Introduction to Python and Jupyter Notebooks
  - Data Cleaning, Bias, and Correlation
  - Best Practices in Visualization and Data Storytelling
  - Interactive Visualization Techniques
  - Predictive Modelling Approaches
- **Practical Phase (Weeks 7-13)**: Bi-weekly meetings to present progress and discuss issues.
- **Final Presentation (Week 14)**: Group presentations of the results and finalization of project reports (with subsequent feedback and grading by the supervisors).
- **Submission at the Data Science Challenge**: The semester's results could be submitted to the Data Science Challenge in the form of a single PDF file (report), a CSV file (predictions), and a ZIP archive (Jupyter notebook file + optional artifacts).

We had provided a Jupyter Notebook as the base for the report as well as a list of recommended literature and an instructor slide deck for a kick-off meeting. Following the course, participants could use instructor feedback to make minor adjustments to their project reports and submit them for the Data Science Challenge. To make sure the students got the most from supervisor feedback at the end of the semester, we moved the submission deadline after the end of the semester.

5 Jury & Evaluation Procedure

Since there was only a relatively short evaluation period of one week (just before the start of the BTW), we had to reduce the evaluation effort for the jury. We achieved this by giving each jury member three submissions, which were to be ranked in relation to each other. Due to overlapping assignments, we were able to create an overall ranking in the end.
The evaluation was done on the basis of criteria mentioned in the challenge description:

- Data Analysis Depth: The thoroughness of your initial data analysis and insights.
- Presentation: The clarity and effectiveness of your data visualization and presentation.
- Model Complexity and Justification: The complexity of your model and the rationale behind choosing it.
- Innovativeness: The novelty of your approach in predicting energy prices.

The review of the last criterion (Accuracy of Predictions: The precision of the model's energy price forecasts.) and the final ranking of the submissions was taken over by the organizers on the basis of the submitted prediction CSV files.

6 Submissions, Presentation & Publication

The call for submissions was followed by 13 teams. Although any participants were allowed in principle, only student groups submitted. Selected contributions were given the opportunity to present their results as a poster / demo at the BTW conference (in the poster and demo reception) and winners got honored at the conference dinner. Given the tight schedule after the submission deadline, we did not opt for publication in the conference proceedings, but rather on the BTW Data Science Challenge website:

https://btw2025.gi.de/data-science-challenge

7 Acknowledgments & Sponsoring

The Data Science Challenge was supported by the BTC AG with prize money totaling 1 000 Euros. The final selection and ranking process would further have not been possible without our awesome jury – a big thank you goes to each and every single one of you:

- Michael Behringer, University of Stuttgart
- Michael Brand, OFFIS
- Stefan Deßloch, University of Kaiserslautern
- Maria Fernanda Davila Restrepo, University of Oldenburg
- Tobias Grimm, Enercon
- Florian Heinz, OTH Regensburg
- Wolfgang Lehner, TU Dresden
- Felix Naumann, HPI
- Martin Poppinga, University of Hamburg
- Tilmann Rabl, HPI
- Mareike Schmidt, University of Hamburg
- Andreas Thor, HTWK Leipzig
- Benjamin Wollmer, Baqend

Autorenverzeichnis