KNOWLEDGE GRAPHS: SEMANTICS, MACHINE LEARNING, AND LANGUAGES

Studies on the Semantic Web

Semantic Web has grown into a mature field of research. Its methods find innovative applications on and off the World Wide Web. Its underlying technologies have significant impact on adjacent fields of research and on industrial applications. This book series reports on the state of the art in foundations, methods, and applications of Semantic Web and its underlying technologies. It is a central forum for the communication of recent developments and comprises research monographs, textbooks and edited volumes on all topics related to the Semantic Web.

Volume 56

Previously published in this series:

Vol. 055 A. Dimou, S. Neumaier, T. Pellegrini and S. Vahdati (Eds.), Towards a Knowledge-Aware AI – SEMANTiCS 2022 – Proceedings of the 18th International Conference on Semantic Systems, 13–15 September 2022, Vienna, Austria
Vol. 054 L. Heling, Decentralized Query Processing over Heterogeneous Sources of Knowledge Graphs
Vol. 053 M. Alam, P. Growth, V. de Boer, T. Pellegrini, H.J. Pandit, E. Montiel, V.R. Doncel, B. McGillivray and A. Meroño-Peñuela (Eds.), Further with Knowledge Graphs – Proceedings of the 17th International Conference on Semantic Systems, 6–9 September 2021, Amsterdam, The Netherlands
Vol. 052 M. Leinberger, Type-Safe Programming for the Semantic Web
Vol. 051 E. Blomqvist, T. Hahmann, K. Hammar, P. Hitzler, R. Hoekstra, R. Mutharaju, M. Poveda-Villalón, C. Shimizu, M.G. Skjæveland, M. Solanki, V. Svátek and L. Zhou (Eds.), Advances in Pattern-Based Ontology Engineering
Vol. 050 M. Mountantonakis, Services for Connecting and Integrating Big Numbers of Linked Datasets
Vol. 049 G. Cota, M. Daquino and G.L. Pozzato (Eds.), Applications and Practices in Ontology Design, Extraction, and Reasoning
Vol. 048 L. Asprino, Engineering Background Knowledge for Social Robots
Vol. 047 Ilaria Tiddi, Freddy Lécué, Pascal Hitzler (Eds.), Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges
Vol. 046 Daniel Dominik Janke, Study on Data Placement Strategies in Distributed RDF Stores
Vol. 045 Pavlos Vougiouklis, Neural Generation of Textual Summaries from Knowledge Base Triples
Vol. 044 Diego Collarana, Strategies and Techniques for Federated Semantic Knowledge Integration and Retrieval

ISSN 1868-1158 (print)
ISSN 2215-0870 (online)

Knowledge Graphs: Semantics, Machine Learning, and Languages

Proceedings of the 19th International Conference on Semantic Systems, 20–22 September 2023, Leipzig, Germany

Edited by

Maribel Acosta

Faculty of Computer Science, Ruhr University Bochum, Germany

Silvio Peroni

Department of Classical Philology and Italian Studies, University of Bologna

Sahar Vahdati

Institute for Applied Informatics, University of Leipzig

Anna-Lisa Gentile

Intelligence Augmentation Group, IBM Research Almaden

Tassilo Pellegrini

University of Applied Sciences St. Poelten, Austria

and

Jan-Christoph Kalo

University of Amsterdam, The Netherlands

IOS Press

AKA

ISBN 978-3-89838-771-2 (AKA, print)
ISBN 978-1-64368-424-6 (IOS Press, print)
ISBN 978-1-64368-425-3 (IOS Press, online)
doi: 10.3233/SSW56

Bibliographic information available from the Katalog der Deutschen Nationalbibliothek (German National Library Catalogue) at https://www.dnb.de

Publisher
Akademische Verlagsgesellschaft AKA GmbH, Berlin

Represented by Co-Publisher IOS Press
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
The Netherlands
Tel: +31 20 688 3355
Fax: +31 20 687 0019
email: order@iospress.nl

PRINTED IN THE NETHERLANDS

Preface

Maribel ACOSTA[a, [0000-0002-1209-2868]], Silvio PERONI[b, [0000-0003-0530-4305]], Sahar VAHDATI[c, [0000-0002-7171-169X]], Anna-Lisa GENTILE[d, [0000-0002-6401-4175]], Tassilo PELLEGRINI[e, [0000-0002-0795-0661]], Jan-Christoph KALO[f,[0000-0002-5492-2292]]

[a] *Faculty of Computer Science, Ruhr University Bochum, Germany, maribel.acosta@rub.de*

[b] *Department of Classical Philology and Italian Studies, University of Bologna, silvio.peroni@unibo.it*

[c] *Institute for Applied Informatics, University of Leipzig, vahdati@infai.org*

[d] *Intelligence Augmentation Group, IBM Research Almaden, annalisa.gentile@ibm.com*

[e] *University of Applied Sciences St. Poelten, Austria, tassilo.pellegrini@fhstp.ac.at*

[f] *University of Amsterdam, The Netherlands, j.c.kalo@uva.nl*

Abstract. This volume encompasses the proceedings of SEMANTiCS 2023, the 19th International Conference on Semantic Systems, a pivotal event for professionals and researchers actively engaged in harnessing the power of semantic computing. At SEMANTiCS, attendees gain a profound understanding of its transformative potential, while also confronting the practical limitations it presents. Each year, the conference magnetizes information managers, IT architects, software engineers, and researchers from a broad spectrum of organizations, spanning research facilities, non-profit entities, public administrations, and the world's largest corporations.

Keywords. Semantic Systems, Knowledge Graphs, Artificial Intelligence, Semantic Web, Linked Data, Machine Learning, Knowledge Discovery

SEMANTiCS serves as a vibrant platform facilitating the exchange of cutting-edge scientific findings in the realm of semantic systems. Furthermore, it extends its scope to encompass novel research challenges in areas such as data science, machine learning, logic programming, content engineering, social computing, and the Semantic Web. Having reached its 19th year, the conference has evolved into a distinguished international event that seamlessly bridges the gap between academia and industry.

Participants and contributors of SEMANTiCS gain invaluable insights from esteemed researchers and industry experts, enabling them to stay abreast of emerging trends and themes within the vast field of semantic computing. The SEMANTiCS community thrives on its diverse composition, attracting professionals with multifaceted roles encompassing artificial intelligence, data science, knowledge discovery and management, big data analytics, e-commerce, enterprise search, technical documentation, document management, business intelligence, and enterprise vocabulary management.

In 2023, the conference embraced the subtitle "Towards Decentralized Knowledge Eco-Systems" and particularly welcomed submissions pertaining to the following topics:

- Web Semantics & Linked (Open) Data
- Enterprise Knowledge Graphs, Graph Data Management
- Machine Learning Techniques for/using Knowledge Graphs (e.g. reinforcement learning, deep learning, data mining and knowledge discovery)
- Knowledge Management (e.g. acquisition, capture, extraction, authoring, integration, publication)
- Terminology, Thesaurus & Ontology Management
- Reasoning, Rules, and Policies
- Natural Language Processing for/using Knowledge Graphs (e.g. entity linking and resolution using target knowledge such as Wikidata and DBpedia, foundation models)
- Crowdsourcing for/using Knowledge Graphs
- Data Quality Management and Assurance
- Mathematical Foundation of Knowledge-aware AI
- Multimodal Knowledge Graphs
- Semantics in Data Science
- Semantics in Blockchain environments
- Trust, Data Privacy, and Security with Semantic Technologies
- Economics of Data, Data Services, and Data Ecosystems
- IoT and Stream Processing
- Conversational AI and Dialogue Systems
- Provenance and Data Change Tracking
- Semantic Interoperability (via mapping, crosswalks, standards, etc.)

Special Sub-Topics:
- Digital Humanities and Cultural Heritage
- LegalTech, AI Safety, Explainable and Interoperable AI
- Decentralized and/or Federated Knowledge Graphs

Application of Semantically Enriched and AI-Based Approaches:
- Knowledge Graphs in Bioinformatics and Medical AI
- Clinical Use Case of AI-based Approaches
- AI for Environmental Challenges
- Semantics in Scholarly Communication and Open Research Knowledge Graphs
- AI and LOD within GLAM (galleries, libraries, archives, and museums) institutions

The Research and Innovation track garnered significant attention with 54 submissions after a call for papers was publicly announced. To ensure meticulous evaluations, an esteemed program committee comprising 85 members collaborated to identify the papers of utmost impact and scientific merit. Implementing a double-blind review process, wherein author identities and the reviewers were obscured to assure anonymity. A minimum of three independent reviews were conducted for each submission. Upon completion of all reviews, the program committee chairs meticulously compared and deliberated on the evaluations, addressing any disparities or differing viewpoints with the reviewers. This comprehensive approach facilitated a meta-review, enabling the committee to recommend acceptance or rejection of each paper. Ultimately, we were pleased to accept 16 papers, resulting in an acceptance rate of 29.6%.

In addition to the peer-reviewed work, the conference had three renowned keynotes from Xin Luna Dong (Meta Reality Lab), Marco Varone (Expert.ai), and Aidan Hogan (Department of Computer Science, University of Chile).

Additionally, the program had posters and demos, a comprehensive set of workshops, as well as talks from industry leaders.

We thank all authors who submitted papers. We particularly thank the program committee which provided careful reviews in a quick turnaround time. Their service is essential for the quality of the conference.

Special thanks also go to our sponsors without whom this event would not be possible:

Gold Sponsors: Metaphacts, Pantopix, PoolParty, TopQuadrant
Silver Sponsors: GNOSS, IOLAR, Ontotext, neo4j, RDFOX, The QA Company
Bronze Sponsor: RWS
Startup Sponsor: Karakun, SP Semantic Partners

Sincerely yours,

The Editors

Leipzig, September 2023

This page intentionally left blank

About the Conference

Peer Review Statement

Research and Innovation track
Submissions received: 54 papers
Accepted: 16 papers (29,6%)
Peer review process: double-blind reviewing process with a minimum of three independent reviews for each submission

Organizing Committee

CONFERENCE CHAIRS
- Sahar Vahdati, Institute for Applied Informatics (InfAI), Dresden, Germany
- Annalisa Gentile, IBM Research, San Jose, CA, USA

RESEARCH AND INNOVATION CHAIRS
- Silvio Peroni, University of Bologna, Bologna, Italy
- Maribel Acosta, Ruhr University Bochum, Bochum, Germany

INDUSTRY CHAIRS
- Christian Dirschl Wolters Kluwer, Munich, Germany
- Marco Brattinga, Ordina, Netherlands
- Thomas Thurner, Semantic Web Company, Vienna, Austria

POSTER AND DEMO CHAIRS
- Sebastian Neumaier, St. Pölten University of Applied Sciences, St. Pölten Austria
- Neha Keshan, Rensselaer Polytechnic Institute, New York, USA

WORKSHOP AND TUTORIALS CHAIRS
- Anisa Rula, University of Brescia, Brescia, Italy
- Jennifer D'Souza, Technische Informationsbibliothek (TIB), Hannover, Germany

LOCAL CHAIRS
- Julia Holze, Institute for Applied Informatics (InfAI), Leipzig, Germany
- Michael Martin, Institute for Applied Informatics (InfAI), Leipzig,Germany

SPONSORING CHAIR
- Martin Kaltenböck, Semantic Web Company, Vienna, Austria

PROCEEDINGS CHAIRS
- Tassilo Pellegrini, St. Pölten University of Applied Sciences, St. Pölten, Austria
- Jan Christoph Kalo, University of Amsterdam, The Netherlands

DIGITAL CONTENT CHAIRS

- Ivan Heibi, University of Bologna, Bologna, Italy
- Daniel Garijo, Universidad Politécnica de Madrid, Madrid, Spain
- Mélisa Machuret, Better Way Devs, Villa De Leyva, Boyacá, Colombia

PUBLICITY CHAIRS

- Angelo Salatino, The Open University, Milton Keynes, UK
- Katharina Koeb, St. Pölten University of Applied Sciences, St. Pölten, Austria

Program Committee Members

Alessandro Adamou	Bibliotheca Hertziana – Max Planck Institute for Art History
Mehwish Alam	Télécom Paris, Institute Polytechnique de Paris
Vito Walter Anelli	Politecnico di Bari
Sören Auer	TIB Leibniz Information Center Science & Technology and University of Hannover
Carlos Badenes-Olmedo	Universidad Politécnica de Madrid
Dhananjay Bhandiwad	InfAI
Stefan Bischof	Siemens AG Österreich
Russa Biswas	Hasso Plattner Institute
Carlos Bobed	University of Zaragoza, Spain
Loris Bozzato	Fondazione Bruno Kessler
Paul Buitelaar	NUI Galway
Pierre-Antoine Champin	LIRIS, Université Claude Bernard Lyon1
Ioannis Chrysakis	FORTH-ICS (Greece) and Dept. of Electronics and Information Systems, Ghent University – IDLab – imec
Fabio Ciotti	Università Roma Tor Vergata
Davide Colla	Department of Historical Studies, University of Turin
Oscar Corcho	Universidad Politécnica de Madrid
Rossana Damiano	Dipartimento di Informatica, Università di Torino
Laura Daniele	TNO
Marilena Daquino	University of Bologna
Anastasia Dimou	Department of Electronics and information systems, University of Ghent
Christian Dirschl	Wolters Kluwer Germany
Milan Dojchinovski	Czech Technical University in Prague
Mauro Dragoni	Fondazione Bruno Kessler – FBK-IRST
Jennifer D'Souza	TIB Leibniz Information Centre for Science and Technology University Library
Victoria Eyharabide	STIH Laboratory, Sorbonne University
Michael Färber	Karlsruhe Institute of Technology
Catherine Faron	Université Côte d'Azur
Said Fathalla	Smart Data Analytics (SDA), University of Bonn, Germany
Agata Filipowska	Department of Information Systems, Poznan University of Economics
Nuno Freire	INESC-ID

Roberto Garcia	Universitat de Lleida
Daniel Garijo	Universidad Politécnica de Madrid
Anna Lisa Gentile	IBM Research
Alasdair Gray	Heriot-Watt University
Ivan Heibi	University of Bologna
Lars Heling	Stardog Union
Nathalie Hernandez	IRIT
Fabian Hoppe	FIZ Karlsruhe, Karlsruhe Institute of Technology
Zhisheng Huang	Vrije Universiteit Amsterdam
Luis Ibanez-Gonzalez	University of Southampton
Shimaa Ibrahim	Bonn University
Antoine Isaac	Europeana & VU University Amsterdam
Marc Jacobs	Fraunhofer
Tobias Käfer	Karlsruhe Institute of Technology
Jan-Christoph Kalo	University of Amsterdam
Naouel Karam	Fraunhofer
Neha Keshan	Rensselaer Polytechnic Institute
Fahad Khan	Istituto di Linguistica Computazionale "Antonio Zampolli"
Dimitris Kontokostas	Medidata
Julius Köpke	Alpen-Adria-Universität Klagenfurt Institute for Informatics Systems
Jasmin Lampert	Austrian Institute of Technology
Maxime Lefrançois	MINES Saint-Etienne
Georgios Lioudakis	ICT abovo P.C.
Sandra Lovrenčić	University of Zagreb, Faculty of organization and informatics Varazdin,
Chiara Lucifora	University of Bologna
Andrea Mannocci	ISTI-CNR
Enrico Mensa	University of Turin, Computer Science Department
Paul Mulholland	The Open University
Sebastian Neumaier	St. Pölten University of Applied Sciences, Austria
Lyndon Nixon	MODUL Technology GmbH
Andrea Giovanni	Nuzzolese University of Bologna
Leo Obrst	MITRE
D. O'Sullivan	Trinity College Dublin
Harshvardhan J. Pandit	ADAPT Centre, School of Computing, Dublin City University
Tassilo Pellegrini	University of Applied Sciences St. Pölten
Jędrzej Potoniec	Poznan University of Technology
Anisa Rula	University of Brescia
Angelo Antonio Salatino	The Open University
Stefan Schlobach	Vrije Universiteit Amsterdam
Pavel Shvaiko	Informatica Trentina
Blerina Spahiu	Università degli Studi di Milano Bicocca
Simon Steyskal	Siemens AG Austria
Tabea Tietz	FIZ Karlsruhe
Sanju Tiwari	Universidad Autonoma de Tamaulipas
Konstantin Todorov	LIRMM / University of Montpellier
Jürgen Umbrich	Vienna University of Economy and Business (WU)

Sahar Vahdati	InfAI
Herbert Van De Sompel	Data Archiving Networked Services
Fabio Vitali	University of Bologna
Joerg Waitelonis	yovisto GmbH
Shenghui Wang	University of Twente
Rigo Wenning	W3C
Wolfram Wöß	Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz

Contents

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230002

Knowledge-Grounded Target Group Language Recognition in Hate Speech

Paula REYERO LOBO [a,1], Enrico DAGA [a] Harith ALANI [a] and Miriam FERNANDEZ [a]

[a] *Knowledge Media Institute, The Open University, United Kingdom*

ORCiD ID: Paula Reyero Lobo https://orcid.org/0000-0001-5238-4550, Enrico Daga https://orcid.org/0000-0002-3184-5407, Harith Alani https://orcid.org/0000-0003-2784-349X, Miriam Fernandez https://orcid.org/0000-0001-5939-4321

Abstract. Hate speech comes in different forms depending on the communities targeted, often based on factors like gender, sexuality, race, or religion. Detecting it online is challenging because existing systems are not accounting for the diversity of hate based on the identity of the target and may be biased towards certain groups, leading to inaccurate results. Current language models perform well in identifying target communities, but only provide a probability that a hate speech text contains references to a particular group. This lack of transparency is problematic because these models learn biases from data annotated by individuals who may not be familiar with the target group. To improve hate speech detection, particularly target group identification, we propose a new hybrid approach that incorporates explicit knowledge about the language used by specific identity groups. We leverage a Knowledge Graph (KG) and adapt it, considering an appropriate level of abstraction, to recognise hate speech-language related to gender and sexual orientation. A thorough quantitative and qualitative evaluation demonstrates that our approach is as effective as state-of-the-art language models while adjusting better to domain and data changes. By grounding the task in explicit knowledge, we can better contextualise the results generated by our proposed approach with the language of the groups most frequently impacted by these technologies. Semantic enrichment helps us examine model outcomes and the training data used for hate speech detection systems, and handle ambiguous cases in human annotations more effectively. Overall, infusing semantic knowledge in hate speech detection is crucial for enhancing understanding of model behaviors and addressing biases derived from training data.

Keywords. hate speech, semantic enrichment, knowledge graphs, language models

1. Introduction

One of the challenges when addressing online hate speech is the extensive use of specialized language that is specific to the communities that are most frequently targeted. A motivating example is illustrated by **Figure 1**. We show two posts from a well-known

[1]Corresponding Author: Paula Reyero Lobo, Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom; E-mail: paula.reyero-lobo@open.ac.uk.

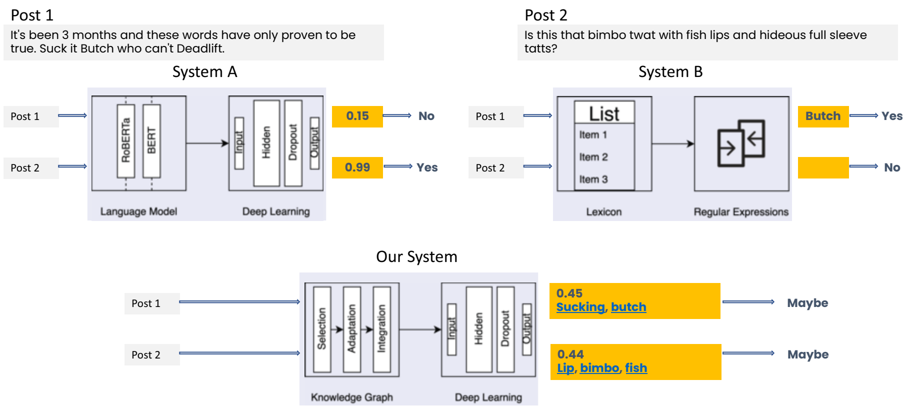


Figure 1. Hate speech recognisers based on language sensitive to gender and sexual orientation. Our approach (**Our System**) embeds a knowledge graph in a deep learning model to give a probability estimate that is competitive with state-of-the-art language models (**System A**), while providing more semantic information supporting the prediction than the existing linguistic approaches of lower accuracy (**System B**).

hate speech corpus [1]. To make appropriate decisions about the hateful nature of these posts, it is crucial to be familiar with the language being used.

Looking at it from the perspective of gender and sexual orientation, certain terms like "butch" and "bimbo" carry connotations related to a woman's masculinity or physical appearance. Depending on the context, these terms can be used to insult or reinforce social stereotypes. These subtle differences in language make the task of recognising hate speech highly dependent on the specific identity groups involved and the context in which the language is used [2].

There are two main approaches to recognise target group language in hate speech. The first, which we name **System A**, are supervised learning approaches [3,4,5] and the state-of-the-art (SOTA) are based on language models [6]. While they can achieve high performance, they only give probabilities of the posts containing references to particular identity groups. Only the second post in our example would contain sensitive language with high probability, but it leaves no further information as to why. This lack of supporting information is concerning due to the subjective interpretations and biases of human annotators. Judgement of hate speech varies significantly according to demographic characteristics [7,8], as it is to be expected that any human annotator will lack familiarity with the language of a particular target. Thus, integrating grounding knowledge may help to better understand predictions, but crucially to make the model more robust to biases in the training datasets. The second, which we name **System B**, are linguistic approaches displaying higher transparency [9,10,11]. They provide references using a list of terms (lexicon) or regular expression patterns and would identify, for example, relevant terminology in the first post [12]. However, they are less accurate as they only capture a sparse representation of the language sensitive to an identity group.

In this work, we aim to integrate a Knowledge Graph (KG) to enrich state-of-the art language model predictions with the entities supporting a decision, while preserving an optimal predictive performance (**Our System**). Following our motivating example, even if the model gives borderline probabilities for the particular posts, the additional semantic information helps to understand the prediction better. Representing terms as entities provides useful semantic relations and properties such as definitions or synonyms, which we can exploit when auditing the model and data, as well as developing the hybrid approach.

Our contributions can be summarised as follows. We propose a conceptual framework to combine semantic knowledge in the form of a KG with an existing deep learning architecture (§3). Specifically, we propose a novel entity weighting scheme to effectively adapt a KG to text classification. We conduct a thorough quantitative and qualitative evaluation of our proposed hybrid learning framework (§4). Particularly, by comparing it against SOTA approaches on recognising language references to gender and sexual orientation in a variety of hate speech datasets. Our proposed semantically enriched approach displays equivalent performance to the use of language models, with transparency and higher generalisability to external datasets. The rest of the paper is structured as follows. (§2) summarises the related work, and (§5) our proposed approach, its strengths and limitations, and concludes the work. The instructions for accessing the data, code for training hybrid models in new domains, or applying them to new data, are published in a public repository[2].

2. Related Work

Content moderation systems generally focus on defining policies to protect any identity group or individual targeted [13]. Nevertheless, the specific sociolinguistic aspects of harmful expressions [14,15] make this phenomenon different for each target. A system focused on recognising hate directed to a specific group would not generalise to a different identity [2]. Similarly, linguistic nuances across group identities significantly impact the annotation of training datasets and lead to inconsistent labelling. Humans who labelled the data have different cultural backgrounds or beliefs [7], are exposed to language sensitive to groups with whom they may not self-identify [16,17], and their subjective interpretations of hate speech differ [18,19], especially when recognising identity groups that are frequent hate targets [20]. Lexical biases, where algorithms associate hate with any language that refers to a particular minority group [10,21], make it critical to analyse hate in terms of the identities targeted.

Due to these issues, one stream of work (**System B**) has been based on the manual selection of terms or expressions to recognise language references to identity groups in hate speech data. To a greater extent, these consist of direct references to members of an identity group (so-called group identifiers, identity terms or identity mentions) [10,12,9], or expressions that comprise potentially offensive language depending on the context, including slurs and objectifying outdated terms, as well as reclaimed slurs [11]. However, these approaches have mainly been used in the development of techniques to mitigate lexical bias in hate speech [22,23], or to measure the effectiveness of such mitigation techniques [24,25,26,27]. Existing approaches based on structured knowledge can only partially cover the prejudice towards identity groups in hate speech training data.

Another prominent line of work (**System A**) relies on supervised learning to more effectively recognise language references to identity groups [4,6,3,5]. These references may refer to broad groups based on sensitive attributes such as gender, race, sexual ori-

[2] https://github.com/preyero/hate-speech-identities

entation, disability or religion; or to the specific affected communities within an identity (e.g. women, transgender, male or other gender subgroups). To the best of our knowledge, these systems may display high performance, but only output a probability to indicate how much language sensitive to a group the post contains. However, in addition to the problems already discussed, language models also acquire biases from the large corpora used in pretraining [28]. Using additional sources of knowledge can produce models that are less flexible and more robust to specific annotation schemes, domain and contexts than deep learning models alone [29].

Semantic knowledge integration has helped to address bias related to the data annotation [30], to generalise better to unseen data [31], and to explain model predictions [32]. These examples imply, on the one hand, that it should be plausible to better overcome the discussed training data specificities given an adequate source of knowledge representation of the language from these communities. On the other, additional knowledge could enrich probabilistic model predictions to better understand them. A common challenge for hybrid approaches remains in finding the correct level of abstraction when applying semantic knowledge to a downstream task [29]. In this work, we propose a novel hybrid approach for grounding deep learning predictions in relevant knowledge for the task (**Our System**). Prior adaptation to the language distribution of the downstream task enables to integrate a KG simply and effectively into the model, without sacrificing predictive performance.

Particularly in hate speech, our approach allows focusing the detection from the target's perspective, as it highlights the specific entities representing language references that influence the prediction. Attending only to the signals learned with standard supervised learning (System A) has shown to acquire annotation and lexical biases from the training data and, in the worst cases, has lead marginalization and censorship of communities at risk [33,34]. Linguistic approaches (System B) to probe these systems for fairness fall short in addressing the language that refers to these communities [25,24]. With the domain specific constraints set by a knowledge-grounded approach (Our System), we intend to bring more focus on the language sensitive to frequent target communities to better understand and prevent online hate.

3. Hybrid Approach

In this section, we present a conceptual framework for supporting text classification with semantic knowledge [3]. First, we present the rationale for selecting semantic knowledge and describe the information leveraged in our approach (§3.1). Second, we present an adaptation phase. The goal is to assign weights to the KG entities based on their relevance in a pretraining corpus from the task domain (§3.2). Finally, we describe the integration of the adapted KG with a deep learning architecture (§3.3). The resulting six hybrid model versions are described in **Table 1**.

3.1. Semantic Knowledge Selection

A Knowledge Graph is a structured representation of knowledge that captures relationships between entities in a particular domain. It is a type of knowledge representation

[3]Note that, while we have selected semantic knowledge to cover specifically language from gender and sexual orientation, this is not fixed in our approach, and the KG can be exchanged.

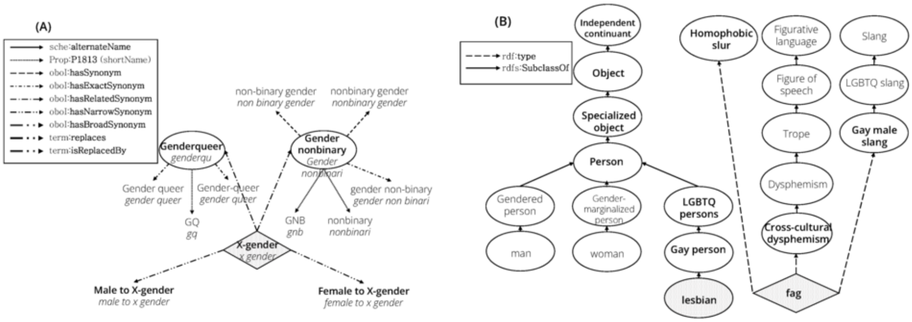


Figure 2. Knowledge Graph class (◯) and instance (◇) entities (*Gender, Sex, and Sexual Orientation* [35]). **(A)** Matching “X-gender” to text. **(B)** Hierarchical entity expansion of “lesbian” and “fag”. Entities in bold are used to match or assign weight to the entities, respectively.

often used in combination with Machine Learning (ML) techniques, as they can help to improve model performance and interpretability in a variety of tasks including search, question answering and natural language understanding [29,36].

To select the particular KG for the task, we explore a wide range of existing KGs that could have comprehensive information about language sensitive to a group, including well-known KGs such as wikidata, DBPedia, and YAGO [37,38,39]. While some of these KGs contain information related to many identities, we selected the Gender, Sex and Sexual Orientation Ontology [35] as a more specific source of knowledge to base hate predictions on the language of a target group. This KG aims to be an integrated vocabulary system to address the lack of standardised gender and sexual orientation terminology in healthcare and includes, to this date, over 16,000 entities and 292 properties. To the best of our knowledge, it is the most comprehensive and up to date to describe these two common hat targets that we can consider for assessing our approach.

We show how we leverage this information following the examples in **Figure** 2. KG entities can capture semantic concepts (e.g., “gay male slang”) or their concrete examples (e.g., “fag”). Additionally, there are object properties to make connections between entities, and data properties to describe the specific values or attributes of entities. For example, the wikidata property `Prop:P1813` indicates that the literal “GQ” can be a short name for the entity “Genderqueer”. We use this richer representation of terms as entities to facilitate the matching of the KG to the texts **(A)**. We use properties like `rdf:type` (to link instances to their classes, such as “fag” to “homophobic slur”) and `rdfs:SubClassOf` (to make hierarchical connections between the classes, e.g., “lesbian” to “gay person”), to exploit the KG structure in our hybrid approach **(B)**.

3.2. *Knowledge Graph Adaptation*

One major challenge when applying semantic knowledge in combination with ML to address particular tasks is the level of abstraction of its information [40]. The existing entities can sometimes encapsulate information that is too abstract or too fine-grained for the task at hand. In this work, we propose an adaptation phase that allows us to weigh the KG entities based on pretraining data. The aim is to give more relevance to those entities that better encapsulate the group language and information, which adds an additional dimension to the factual and structural information of previous hybrid approaches [41].

3.2.1. Search for Pretraining Data

To learn the weights of the KG we selected a balanced subset of the Jigsaw Toxicity dataset. For a full description of the datasets used in this work for training and validation, including references, descriptions, and statistics, see (§4.1) and **Table 2**. To create the *Jigsaw Sample* we selected all the texts annotated with sexual orientation, a total of 12,713 texts, and a random same size set that includes 16,850 texts annotated as related to gender. We then built a stratified sample of texts from all the remaining identities (i.e., religion, race, disability, and none) as the negative class. This provided us a balanced dataset of 50,852 texts, with 50% of them related to the class, and 50% related to any other identities.

3.2.2. Entity recognition

To determine whether an entity appears or not in a given text (*entity matching*) we take into account, not just the entity's label, but also its alternative names and existing synonyms. For example, in **Figure 2 (A)** we observe that "X-gender" is defined by "Male to X-gender","Female to X-gender", "Genderqueer" and "Gender nonbinary". In addition to these terms, we also consider stemming variations, such as "genderqu" and "gender nonbinari". Any text that contains any of these expressions is considered a text where the entity appears. Specifically, the KG properties shown in **Figure 2 (A)** are used to derive the synonyms for the entity matching. We develop a search index based on the Whoosh 2.7.4 library (https://whoosh.readthedocs.io) to speed up the entity matching, and obtain the stemming variations with its Porter stemmer native function.

3.2.3. Entity weighting

Finally, we consider two types of entity weighting schemes: (i) based on the frequencies of entities in the pretraining data and (ii) based on the learned coefficients of a ML model used for the domain task, in this case, the binary classification.

Entity weighting based on frequency (**DocF**) The weight provided to the entities is based on the ratio of appearance of that entity within the positive sample (i.e., all the texts related to gender and sexual orientation) vs. the negative sample (i.e., all the text related to any other identity). Lets D_p be the set of all texts related to the class, and D_n be the set of all text related to any other identity. Given an entity e_i, we consider the occurrences e_i in D_p (D'_p), and the occurrences e_i in D_n (D'_n). The weight of the entity $w(e_i)$ is then computed as $w(e_i) = D'_p/D_p - D'_n/D_n$, such that $w \in$ [-1, 1].

Entity weighting based on ML coefficients (**LR** and **MultiNB**) This approach provides weights to the different entities based on the coefficients defined by a machine learning model. The coefficients reflect how discriminative the entities are when predicting whether a particular text refers (or not) to the class. We use two different ML models for the task: Logistic Regression (LR) and Multinomial Naive Bayes (MultiNB). As input to train the ML models, we provide for each pretraining sample: (i) a class label (i.e., whether the text contains any language references to gender and sexual orientation) and (ii) the one-hot-encoding of the entities resulting from the *entity matching* explained above. That is, the ML models use entities as features for the classification. The resulting coefficients reflect the feature importance and how much each entity contributes to the prediction.

Additional modification of the weighing schemes based on hierarchical entity expansion To test whether the KG structure could serve us to better refine the adaptation, we propose a modification affecting the entities to be included in the weighting process. **Figure 2 (B)** shows in bold the process of expanding both an entity that is class (e.g., "lesbian") or an instance (e.g., "fag"). Every class expands up to its top-level using the `rdfs:SubclassOf` property to gather, e.g., that "lesbian" is a Gay Person, LGBTQ person, Person, and so on. For every instance, we would expand based on the `rdf:type` property and also include that "fag" is a Homophobic slur, Cross-cultural dysphemism, and Gay male slang, for example. Thus, in this modified version of the weighting scheme, an entity e_i is considered mentioned in a text if the entity e_i itself, any of its subclasses, or any of its instances appears in the text.

3.3. Knowledge Graph Integration

This phase describes how the adapted KG can be embedded with a deep learning architecture. In the following, we present the two main components of the proposed hybrid learning framework.

Semantic component Our hybrid approach considers an adapted KG (KG with pretraining weights) in the feature extraction. The weights of the entities found in the training samples constitute the feature vectors that are used as inputs. That is, the input for the deep learning component is a sparse vector representation, where the non-zero components are the weights of the entities in the training samples. Compared to contextualised word embeddings, the KG-based feature extraction provides a lower dimensional numerical representation of the input texts. We compare our hybrid approach to pretrained transformer architectures used in the SOTA, where RoBERTa [42] is the best-performing as compared to BERT [43] and the Universal Sentence Encoder [44].

Deep learning component The deep learning architecture used in the SOTA for recognising target group language in hate speech consists of a Feed-foward Multilayer Neural Network with a dropout layer and M binary layers for classification, one for each group identity (gender, sexual orientation, religion, race, disability, national origin, and age) [6]. That is, for a given input text, the model provides M probabilities indicating whether it contains any language related to each group. Because our work is focused on gender and sexual orientation, we only consider the probability of belonging to any of these two classes.

As a result, we obtain the six different hybrid model versions described in **Table 1.** The hyperparameters are the same for training all models, using 8 as the size of the training batches, the number of hidden layers set to 256, and a 0.05 dropout rate in the Feed-forward neural layer.

Table 1. Hybrid models (in bold) resulting from the different adaptation schemes (§3.2) used for hybrid feature extraction. H.E indicates the model variation when including hierarchical entity expansion.

Version	Description	H.E
HybridDocF	Features based on the ratio of entity occurrences	**HybridDocF_h**
HybridLR	Features based on coefficients of entities in a linear regression	**HybridLR_h**
HybridMultiNB	Features based on the coefficients of entities in a multinomial Naive Bayes model	**HybridMultiNB_h**

Table 2. Number and (%) of texts that are related to each identity group in training and validation datasets. A text may relate to none, one or more identity groups.

Identity	Jigsaw	Jigsaw_{Sample}	Measuring Hate Speech	Gab Hate Corpus*	HateXplain	$\text{XtremeSpeech}_{English}$
Gender	88790(19.82%)	16850(33.14%)	14825(37.47%)	568(7.27%)	1375(11.15%)	145(5.49%)
S. Orientation	12713(2.84%)	12713(25.00%)	7719(19.51%)	355(4.54%)	1643(13.32%)	39(1.48%)
Religion	70149(15.66%)	12683(24.94%)	6578(16.63%)	1347(17.24%)	3781(30.66%)	79(2.99%)
Race	42906(9.58%)	9674(19.02%)	12635(31.93%)	1711(21.90%)	4597(37.27%)	34(1.29%)
Disability	5559(1.24%)	4918(9.67%)	1120(2.83%)	241(3.08%)	54(0.44%)	
Origin			7744(19.57%)	1202(15.38%)	642(5.21%)	
Economic					9(0.07%)	23(0.87%)
Age			1051(2.66%)			
Politics				3063(39.2%)		
Any other					712(5.77%)	701(26.56%)
Total	448000	50852	39565	7813	12334	2639

4. Evaluation

In this section, we present our evaluation setup (§4.1) as well as the quantitative results against SOTA approaches for recognising language sensitive to gender and sexual orientation in hate speech (§4.2). (§4.3) provides the results from our qualitative evaluation. Specifically, an error analysis of the best-performing hybrid model (§4.3.1) and a data and model prediction analysis guided by the KG (§4.3.2).

4.1. Experimental Setup

This section describes the datasets used for training and testing our proposed hybrid models, and the baselines and metrics used for evaluation.

4.1.1. Data

We consider five datasets for training and testing our models. See **Table 2** for specific statistics and data descriptions.

Jigsaw [45]: To the best of our knowledge, this is the largest public toxicity corpus containing annotations of identity groups, with 448k annotated posts from the Civil Comments platform. These texts are annotated with a binary indicator of toxicity (toxic/non-toxic) and with the identity groups mentioned in them. Group annotations are based on the following identities: gender, sexual orientation, race, religion, disability or no mention of an identity group.

Measuring Hate Speech [20]: This dataset constitutes the largest hate speech training corpus and was used in the SOTA [6]. It contains 39,565 texts collected from Reddit, Youtube, and Twitter, and annotated with gender, sexual orientation, race, religion, age, disability and national origin identities. The gender and sexual orientation categories constitute 56.98% of the dataset.

Gab Hate Corpus [1]: This commonly used dataset contains 7813 texts collected from Gab, which were deemed hateful by the annotators and provide additional annotations for gender, sex, race, religion, disability, and political ideology.

XtremeSpeech English [46]: The complete dataset contains 5,180 texts collected from Facebook, Twitter and WhatsApp. The dataset is not yet public, but the authors have kindly shared with us a subset of 2,639 texts written in English that focuses on Kenya as a geographic location. These texts contain dangerous, derogatory and exclusionary speech and are annotated considering the following identities: gender, sexual orientation, religion, race, and economic status.

HateXplain [47]: One of the first datasets that included annotations for identity groups. The corpus provides 12,334 texts collected from Twitter and Gab, and those deemed hateful provide annotations for gender, sexual orientation, race, religion, and national origin identity groups.

We use a subset the Jigsaw dataset as pretraining data for the KG adaptation (see §3.2). The hybrid models (§3.3) and the RoBERTa_base baseline are trained using the *Measuring Hate Speech* corpus with the same data preparation used by [6], and evaluated using the *HateXplain*, *XtremeSpeech English* and *Gab Hate Corpus*. While soft labels are used for training the models (i.e., the proportion of annotations for each text), majority voting is considered in the validation datasets for consistency with the baseline evaluation.

4.1.2. Baselines

We select the most representative System A (supervised learning) and System B (linguistic) approaches as baselines. As System A, a **RoBERTa_base** [6] model sets the upper bound in terms of performance. However, this model does not provide any insights on why texts are associated with a particular identity group and only learns from the training data. As System B, **Toxic Debias** [11] is the list of terms and regular expressions most commonly used for the identification of texts containing sensitive language towards minoritized groups in hate speech. From the 53 potentially offensive and 26 non-offensive mentions to these groups, 47 expressions refer to gender and sexual orientation. We highlight the 14 non-offensive and 33 possibly offensive mentions in our publicly available repository.

4.1.3. Evaluation Metrics

For comparability issues, we adopt the same evaluation as in the supervised learning baseline and consider Accuracy and F1 scores according to a 0.5 threshold, and two threshold-agnostic metrics: the Area under the ROC Curve (ROC AUC) and Area under the Precision-Recall Curve (PR AUC).

4.2. Evaluation Results

This section reports on the effectiveness of our hybrid approach for recognising language sensitive to gender and sexual orientation identities in hate speech datasets. First, we compare hybrid models against the best-performing baseline (System A) with a 5-fold cross-validation (**Figure 3**) for comparability with the original paper [6]. Second, we test the robustness of the linguistic, supervised, and hybrid learning approaches to different data contexts with a thorough evaluation on datasets external to training (**Table 3**). We include, to the best of our knowledge, all published datasets on hate speech that have consider identity groups in their annotation. We note that the linguistic baseline (System B) does not require training. For simplicity, we only include in the table the hybrid models with hierarchical entity expansion as they are the best-performing ones.

Finding 1. *Our proposed hybrid approach is simple and effective, displaying a comparable performance to the SOTA supervised learning (System A) approaches based on language models.*

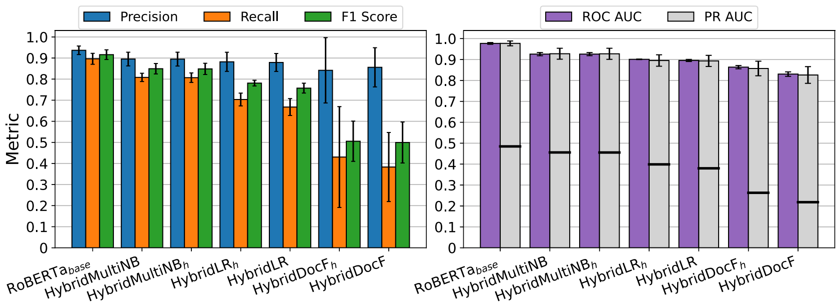


Figure 3. Supervised and hybrid learning model cross-validation results over the training corpus (Measuring Hate Speech). *ML-based hybrid models can be as effective as language models in recognising language references to gender and sexual orientation in hate speech.*

Table 3. Results of the linguistic, supervised learning and hybrid models when testing out of training domain (Gab Hate Corpus, XtremeSpeech$_{English}$, and HateXplain). *Semantic knowledge makes the model more robust to changes in domain and context.*

	Gab Hate Corpus				XtremeSpeech$_{English}$				HateXplain			
Model	Accuracy	$F1$	ROC AUC	PR AUC	Accuracy	$F1$	ROC AUC	PR AUC	Accuracy	$F1$	ROC AUC	PR AUC
Toxic Debias	**91.81**	58.82	74.82	40.20	**94.01**	52.41	72.96	31.12	84.43	67.36	79.01	52.66
HybridDocF$_h$	91.30	51.15	84.52	54.55	93.97	53.91	87.02	47.05	79.45	43.55	78.37	55.96
HybridLR$_h$	90.64	**62.42**	89.30	64.38	90.79	49.27	88.36	50.79	83.48	67.72	88.15	68.35
HybridMultiNB$_h$	89.36	61.11	90.13	68.24	90.38	47.74	87.26	51.80	85.63	73.57	91.38	78.37
RoBERTa$_{base}$	88.85	61.55	**93.06**	**70.32**	92.99	**57.67**	**93.67**	**57.38**	**89.91**	**80.22**	**95.60**	**86.46**

As seen in **Figure 3**, the hybrid models based on ML coefficients (LR and MultiNB) obtain competitive results with a RoBERTa_base model. They outperform the frequency-based models (DocF), particularly in terms of Recall and F1 Score, with lower standard variation across folds. The differences in incidence rates (horizontal black lines of the PR AUC bar to show the proportion of positive predictions across folds) indicate that the HybridDocF predictions are less aligned with the transformer and other hybrid-based models. The figure also shows that hierarchical entity expansion outperforms their counterparts for LR and DocF models, and remains the same in the MultiNB setting.

Finding 2. *The hybrid models display higher generalisability when applied to external datasets than baseline approaches.*

Table 3 shows the generalisability to external validation datasets. As expected, performance drops when evaluating these models with data of a different nature to that used during training (see §4.1.1 and **Table 2** for details on the platform sources, data characteristics and distribution in annotations). This is true, especially in the XtremeSpeech corpus, which captures data from Kenya and English is used in combination with Swahili for some texts. We show, however, that the generalisability of our hybrid models is higher than the baseline, since the gap with the upper bound set by the language models drops with respect to the in-domain evaluation. Aside from enhancing transparency, the introduction of semantic knowledge is key to making these models more robust to context, data and domain changes.

Finding 3. *Our hybrid models display higher performance than the linguistic (System B) baseline while also providing higher levels of interpretability.*

All our proposed hybrid methods outperform Toxic Debias in all metrics except Accuracy in *XtremeSpeech English* and *Gab Hate Corpus*. This is due to imbalanced dataset

Table 4. Error analysis in a False Positives (FP) and False Negatives (FN) sample. A.E indicates the categories that are associated with possible Annotation Errors. N indicates the number of errors found in our sample. *Semantic knowledge provides a better understanding of training data and model outcomes.*

Category (FP)	Definition	A.E	N
Demographic descriptor	Direct explicit reference to a member of the identity group.	X	**117**
Targeted language	Insults, sexually explicit or topics related to the group.	X	20
Implicit reference	Referes to a group member using pronouns.	X	10
False match	Incorrectly flagged due to polysemy.		3

Category (FN)	Definition	A.E	N
No reference	No language related to the group	X	26
Missed at content	Not identified at validation, due to misspellings or being out of training domain.		19
Missed by method	Mention not correctly found or given importance by model.		**85**

conditions. As shown in **Table 2**, these datasets have a lower number of texts from the positive class. System B Accuracy drops below both supervised and hybrid approaches when the proportion of true positives is higher (*HateXplain*), where a model predicting only one class would have a lower chance of obtaining high scores. We observe how performance is significantly higher for the hybrid models in all other evaluation metrics.

In addition to outperforming the linguistic baseline, our hybrid approach provides a higher level of interpretability. While the lexicon only provides terms recognised in the text to categorise it as being associated with the group (e.g., the term "fag"), our hybrid methods provide entities, and with them, their semantic structure. In **Figure 2** we see that, in addition to the label, the KG structure informs about the fact that it is a Gay male slang, a Homophobic slur, and holds different meanings across cultures (i.e., a cross-cultural dysphemism). Similarly, the properties in the KG would also inform that "faggot" is a related synonym, and that it can be replaced by "gay man". KG properties and relations provide a much richer level of knowledge representation than simple terms. This richer source of semantic knowledge has helped to achieve a competitive hybrid baseline with the one based on language models.

4.3. *Exploiting Semantic Enrichment*

This section highlights that the KG is also instrumental to enhance the model's transparency and robustness to problems in hate speech training datasets. We begin our qualitative evaluation with an in-depth error analysis (§4.3.1), and extend it to audit how the training datasets capture language related to these groups (§4.3.2).

4.3.1. *Error Analysis*

Using a thematic analysis approach [48], we identify emerging typologies of errors and group them into distinct categories (**Table 4**). We focus on LR-based hybrid model with hierarchical entity expansion (HybridLR_h) because it outperforms the MultiNB-based models in two of the three validation datasets (**Table 3**). We translate errors into distinct categories considering: (i) each text, (ii) its group identity annotation, (iii) the model predicted probability for the gender and sexual orientation identities and, (iv) the list of entities provided by the model ranked by their weight. Our analysis consists of a 100-quartile random error sample in the validation datasets, to cover equally errors in all ranges of predicted probabilities. Sampling in each of the 3 validation datasets results in 280 texts.

Finding 3. *The semantically enriched predictions provided by the adapted KG enhance the transparency of the model, which helps to better understand model errors and to detect possible annotation errors.*

As shown in **Table 4**, we identified seven distinct categories of errors. We first analysed the false positives (FP) errors (i.e., where the model indicated that the texts mention gender and sexual orientation identities but annotators indicated the opposite). Our analysis reveals that, while most of these texts were not annotated as related, they contain relevant entities, including (i) *demographic descriptors* such as woman, man, girls, male, females, trans, girlfriend, homosexual, gay or lesbian, (ii) *targeted language*, such as insults and sexually explicit references (e.g., sexual assault), and/or (iii) thematically related entities, like birth, feminist, or lgbt. Less distinctive cases include the use of *implicit references* such as pronouns to refer to members of the group. It is important to highlight that all of these instances could be interpreted as annotation errors rather than model errors since the annotators may have missed relevant information in the text. While more experiments are needed to provide robust conclusions, these results seem to indicate that the information provided by the KG could be key to further investigating annotation disagreements. We find only 3 examples that are incorrectly categorised by the model due to polysemy (e.g., "straight" not meaning a sexual orientation), which were clear model errors.

We then analysed the false negative (FN) errors (i.e., instances where our model said that the text did not belong to the categories of gender and sexual orientation and the annotators indicated the opposite). The first category identified is *No Reference*. These are texts that do not display any term associated with sex and sexual orientation. These can also constitute annotation errors, where the annotator wrongly associated the text with these identities. The second category identified is *Missed at content*. These are errors where KG had the relevant entities, but they were not recognised within the text due to spelling mistakes (e.g., "feminisium", "gayfagsex"), or because those entities did not appear in the training corpus used during the KG adaptation phase (e.g., "sexism"). The third category identified is *Missed by method*. This reflects errors where either the KG did not contain the relevant information due to lack of coverage (e.g., "gayzors, "lezbos", "fellatiate", "madam", or "negress") or the relevant entities had a low weight assigned during the KG adaptation phase (e.g., "transphobe", "prostitutes", or "polygamy"). These issues constitute 65% of the errors. In some cases, entities related to the group receive a low weight during KG adaptation due to having noisy synonyms (e.g., "t word" as related synonym of "tranny" and found in texts with "t* word"). These observations could help to improve the specificity of the KG by revisiting which properties to use as synonyms for the entity recognition. Similarly, lack of coverage can also be due to an insufficient level of granularity with the KG (e.g., "daughter" and "son" as synonyms for "child", which is not expressing gender). These insights provide relevant information for improving both the KG and the proposed hybrid solutions.

Overall, we are able to identify these error categories guided by the additional semantic information provided by the hybrid approach. The issues identified along the model pipeline will help us in our future work to refine our hybrid models and enhance its performance. We also draw attention to the finding that grounding predictions on knowledge can help us to better understanding not only model errors, but the ambiguous cases that exist within the data that may be harder to classify by human annotators.

Table 5. KG entities sorted by feature importance that represent the language related to Gender and Sexual orientation in a sample of true predictions and errors. *Semantic knowledge displays hard-to-classify cases for the model and the human annotators.*

Sample	Target Group Language
True Positives	woman, man, LGBTQ, LGBT, .lgbt, man who has sex with men, r/lgbt, male gender identity, lesbian woman, female gender identity, gay man, .gay, transgender person, Black man, gay, homo, gay person, gender, heterosexual, homosexuality, feminist, lesbian, asexual and homoromantic person, gai, A-Gay, heterosexual person, gay identity, human homosexuality, sak veng (long hair), queer sexual orientation, transgender, same-gender marriage, marriage, transgenderism, heterosexuality, womanism, pederasty, lesbian identity, sexuality, heterosexual identity, lesbianism, homosexualism, personal identity, homophobia, queer identity, person who menstruates, mixed-orientation marriage, single person, *sex, feminism, partner, marital partner, sex worker, fag, faggot, masculism, pussy, hers, thot, rape, menstruation, bitch*
Missing in annotation	woman, man, LGBT, woman of color, .lgbt, man who has sex with men, r/lgbt, male gender identity, female gender identity, .gay, gay, gay person, heterosexual, homosexuality, feminist, lesbian, asexual and homoromantic person, gai, A-Gay, gay identity, human homosexuality, queer sexual orientation, transgender, same-gender marriage, marriage, interpersonal orientation, womanism, lesbian identity, sexuality, homosexualism, personal identity, lesbianism, homophobia, queer identity, single person, abusive person, sex, interpersonal attraction, *partner, faggot, semen, pussy, hers, bitch*
Missing in prediction	man who has sex with men, feminist, marriage, homophobia, person who menstruates, sex, **interracial marriage**, **sex work client**, *marital partner,* ***parent****, sex worker, fag, faggot,* ***rapist****,* ***female gender role****, pussy,* ***abortion****,* ***morphological enlargement****, hers,* ***vagina****, thot,* ***penis****, rape, domestic violence,* ***she****,* ***bimbo****,* ***sexual abstinence****, cunt, bitch,* ***he****,* ***whore****,* ***slut****,* ***fuck****,* ***Mrs.****,* ***rainbow flag***

4.3.2. Auditing Training Datasets

Motivated by our error analysis, we exploit our semantically enriched method to assess how hate speech training dataset captures identity group language. The result of this analysis can be seen in **Table 5**. We follow the same approach in (§4.3.1) and draw a 100-quartile sample of true predictions, which includes 286 texts. We then use the elbow method [49] to filter those entities that are more relevant considering the weights provided by the HybridLR_h model. Within this category (*True Positives*) we show those relevant entities extracted from texts where both, the model and the human annotators, agreed that the text was related to these identities. We note that these lists are not intended to provide an exhaustive list of all the language related to gender and sexual orientation. Nevertheless, they provide the minimum set of KG entities required to identify language references to these identities, and gives valuable insights to better understand how common hate targets are captured by the hate speech training datasets.

We conduct the same analysis in the samples with mismatches of annotation and model predictions. Within the category *Missing in annotation*, we display the relevant entities in texts indicated as related by the model, but not by the human annotators. Using the same data from the error analysis, the list includes language from 147 texts, and highlights relevant entities that the annotators may have missed when assessing the texts.

Within the category *Missing in prediction*, we display the relevant entities in texts indicated as related by the human annotators, but not by the model. The list highlights relevant entities that, while available within the KG, were not given enough relevance during the hybrid approach. Entities in italics correspond to those not included by the elbow point due to having a lower weight, but required to identify related texts. Entities in bold are unique to the texts missed in the prediction. This important entities highlight the complexity of learning language, as some of these entities may only be related to the gender and sexual orientation identities in specific contexts (e.g, f*ck as a swear word, or being sexually explicit). The same is true for entities that appear in texts that are only sometimes annotated as related (e.g. woman, LGBT, gay).

Finding 4. *A knowledge-grounded approach for understanding hate from the perspective of gender and sexual orientation identities helps to identify language relevant for their recognition in hate speech, as well as the terminology that may be associated with either model or annotation errors.*

5. Conclusion

We present a novel hybrid approach for grounding deep learning predictions in semantic knowledge relevant to the recognition of language references to gender and sexual orientation in hate speech. First, selecting a KG as semantic knowledge is a richer form of structured knowledge than existing linguistic approaches, providing novel semantically enriched predictions that are as effective to the use of black-box language models. Second, an adaptation phase based on machine learning allows finding an optimal representation level, which is a major challenge for applying semantic knowledge to downstream tasks. Finally, we propose a simple and effective feature-based approach to integrate the adapted KG to a neural network. Our evaluation on gender and sexual orientation demonstrates that a knowledge-grounded approach is key to enhance model transparency, robustness, and handling of annotation errors. Particularly, as it can highlight vocabulary for better understanding how training data captures identity group language, what are the type of errors in the model and, more interestingly, the ambiguities in human annotations.

We acknowledge we only evaluate our approach on two particular groups and one KG. Further research on other target groups would underline the value of knowledge-based approaches to hate speech detection. Similarly, considering a variety of KG domains and sizes would provide valuable insights on how to integrate them more effectively. KGs are generally costly to generate and maintain, and sometimes their coverage may not be sufficient for the task [50]. Our work however shows that, when this knowledge is available, it can positive complement and enhance a standard deep learning approach. We acknowledge the limitations of hate speech evaluation using standard performance metrics and leave as future work settings specific to the task [51,52] tailored to these identities. In terms of annotation findings, our semantically enriched models uncover references in 97% false positive errors. A more exhaustive analysis is needed to investigate the reasons behind these disagreements and the extent to which these cases constitute difficult to classify training examples that could improve hate speech recognisers [53]. Nevertheless, analysing hate in terms of the groups targeted is critical due to the subtlety of this language, which makes the recognition of hate speech even more difficult for annotators to understand and perceive [20].

To conclude, we particularly emphasise that this work does not aim to infer an individual's sensitive attributes [54]. This work rather aims to attend to the sociolinguistic aspects in hate speech in the hope of better contextualising automatic recognition systems with the language use of the social realities they imply.

References

[1] Kennedy B, Atari M, Davani AM, Yeh L, Omrani A, Kim Y, et al. Introducing the Gab Hate Corpus: defining and applying hate-based rhetoric to social media posts at scale. Language Resources and Evaluation. 2022:1-30.

[2] Yoder M, Ng L, Brown DW, Carley K. How Hate Speech Varies by Target Identity: A Computational Analysis. In: Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL). Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics; 2022. p. 27-39. Available from: https://aclanthology.org/2022.conll-1.3.

[3] Silva L, Mondal M, Correa D, Benevenuto F, Weber I. Analyzing the targets of hate in online social media. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 10; 2016. p. 687-90.

[4] Mossie Z, Wang JH. Vulnerable community identification using hate speech detection on social media. Information Processing & Management. 2020;57(3):102087. Available from: https://www.sciencedirect.com/science/article/pii/S0306457318310902.

[5] Waseem Z, Hovy D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: Proceedings of the NAACL Student Research Workshop. San Diego, California: Association for Computational Linguistics; 2016. p. 88-93. Available from: https://aclanthology.org/N16-2013.

[6] Sachdeva P, Barreto R, Von Vacano C, Kennedy C. Targeted Identity Group Prediction in Hate Speech Corpora. In: Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH). Seattle, Washington (Hybrid): Association for Computational Linguistics; 2022. p. 231-44. Available from: https://aclanthology.org/2022.woah-1.22.

[7] Sap M, Swayamdipta S, Vianna L, Zhou X, Choi Y, Smith NA. Annotators with Attitudes: How Annotator Beliefs And Identities Bias Toxic Language Detection. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, United States: Association for Computational Linguistics; 2022. p. 5884-906. Available from: https://aclanthology.org/2022.naacl-main.431.

[8] Gubitz SR. Race, Gender, and the Politics of Incivility: How Identity Moderates Perceptions of Uncivil Discourse. Journal of Race, Ethnicity, and Politics. 2022;7(3):526–543.

[9] Nozza D, Bianchi F, Lauscher A, Hovy D. Measuring Harmful Sentence Completion in Language Models for LGBTQIA+ Individuals. In: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion. Dublin, Ireland: Association for Computational Linguistics; 2022. p. 26-34. Available from: https://aclanthology.org/2022.ltedi-1.4.

[10] Dixon L, Li J, Sorensen J, Thain N, Vasserman L. Measuring and Mitigating Unintended Bias in Text Classification. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. AIES '18. New York, NY, USA: Association for Computing Machinery; 2018. p. 67–73. Available from: https://doi.org/10.1145/3278721.3278729.

[11] Zhou X, Sap M, Swayamdipta S, Choi Y, Smith N. Challenges in Automated Debiasing for Toxic Language Detection. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Available from: https://par.nsf.gov/biblio/10308662.

[12] Smith EM, Hall M, Kambadur M, Presani E, Williams A. "I'm sorry to hear that": Finding New Biases in Language Models with a Holistic Descriptor Dataset. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics; 2022. p. 9180-211. Available from: https://aclanthology.org/2022.emnlp-main.625.

[13] Calabrese A, Ross B, Lapata M. Explainable Abuse Detection as Intent Classification and Slot Filling. Transactions of the Association for Computational Linguistics. 2022 12;10:1440-54. Available from: https://doi.org/10.1162/tacl_a_00527.

[14] Saha K, Kim SC, Reddy MD, Carter AJ, Sharma E, Haimson OL, et al. The Language of LGBTQ+ Minority Stress Experiences on Social Media. Proc ACM Hum-Comput Interact. 2019 nov;3(CSCW). Available from: https://doi-org.libezproxy.open.ac.uk/10.1145/3361108.

[15] Kwarteng J, Perfumi SC, Farrell T, Third A, Fernandez M. Misogynoir: challenges in detecting intersectional hate. Social Network Analysis and Mining. 2022;12(1):166.

[16] Goyal N, Kivlichan ID, Rosen R, Vasserman L. Is Your Toxicity My Toxicity? Exploring the Impact of Rater Identity on Toxicity Annotation. Proc ACM Hum-Comput Interact. 2022 nov;6(CSCW2). Available from: https://doi-org.libezproxy.open.ac.uk/10.1145/3555088.

[17] Mastromattei M, Basile V, Zanzotto FM. Change My Mind: How Syntax-based Hate Speech Recognizer Can Uncover Hidden Motivations Based on Different Viewpoints. In: Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022. Marseille, France: European Lan-

guage Resources Association; 2022. p. 117-25. Available from: https://aclanthology.org/2022.nlperspectives-1.15.

[18] Rottger P, Vidgen B, Hovy D, Pierrehumbert J. Two Contrasting Data Annotation Paradigms for Subjective NLP Tasks. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, United States: Association for Computational Linguistics; 2022. p. 175-90. Available from: https://aclanthology.org/2022.naacl-main.13.

[19] Kazienko P, Bielaniewicz J, Gruza M, Kanclerz K, Karanowski K, Miłkowski P, et al. Human-centered neural reasoning for subjective content processing: Hate speech, emotions, and humor. Information Fusion. 2023;94:43-65. Available from: https://www.sciencedirect.com/science/article/pii/S1566253523000167.

[20] Sachdeva P, Barreto R, Bacon G, Sahn A, von Vacano C, Kennedy C. The Measuring Hate Speech Corpus: Leveraging Rasch Measurement Theory for Data Perspectivism. In: Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022. Marseille, France: European Language Resources Association; 2022. p. 83-94. Available from: https://aclanthology.org/2022.nlperspectives-1.11.

[21] Xu A, Pathak E, Wallace E, Gururangan S, Sap M, Klein D. Detoxifying Language Models Risks Marginalizing Minority Voices. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics; 2021. p. 2390-7. Available from: https://aclanthology.org/2021.naacl-main.190.

[22] Kennedy B, Jin X, Mostafazadeh Davani A, Dehghani M, Ren X. Contextualizing Hate Speech Classifiers with Post-hoc Explanation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 5435-42. Available from: https://aclanthology.org/2020.acl-main.483.

[23] Zhang G, Bai B, Zhang J, Bai K, Zhu C, Zhao T. Demographics Should Not Be the Reason of Toxicity: Mitigating Discrimination in Text Classifications with Instance Weighting. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 4134-45. Available from: https://aclanthology.org/2020.acl-main.380.

[24] Attanasio G, Nozza D, Hovy D, Baralis E. Entropy-based Attention Regularization Frees Unintended Bias Mitigation from Lists. In: Findings of the Association for Computational Linguistics: ACL 2022. Dublin, Ireland: Association for Computational Linguistics; 2022. p. 1105-19. Available from: https://aclanthology.org/2022.findings-acl.88.

[25] Cai Y, Zimek A, Wunder G, Ntoutsi E. Power of Explanations: Towards automatic debiasing in hate speech detection. arXiv e-prints. 2022 Sep:arXiv:2209.09975.

[26] Sen I, Samory M, Wagner C, Augenstein I. Counterfactually Augmented Data and Unintended Bias: The Case of Sexism and Hate Speech Detection. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, United States: Association for Computational Linguistics; 2022. p. 4716-26. Available from: https://aclanthology.org/2022.naacl-main.347.

[27] Chuang YS, Gao M, Luo H, Glass J, Lee Hy, Chen YN, et al. Mitigating Biases in Toxic Language Detection through Invariant Rationalization. In: Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021). Online: Association for Computational Linguistics; 2021. p. 114-20. Available from: https://aclanthology.org/2021.woah-1.12.

[28] Schramowski P, Turan C, Andersen N, Rothkopf CA, Kersting K. Large pre-trained language models contain human-like biases of what is right and wrong to do. Nature Machine Intelligence. 2022;4(3):258-68.

[29] Futia G, Vetrò A. On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research. Information. 2020 Feb;11(2):122. Available from: http://dx.doi.org/10.3390/info11020122.

[30] Reyero Lobo P, Daga E, Alani H, Fernandez M. Semantic Web technologies and bias in artificial intelligence: A systematic literature review. Semantic Web. 2023;14(4):745-70. Publisher: IOS Press.

[31] Cui L, Wu Y, Liu S, Zhang Y. Knowledge Enhanced Fine-Tuning for Better Handling Unseen Entities in Dialogue Generation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational

Linguistics; 2021. p. 2328-37. Available from: https://aclanthology.org/2021.emnlp-main.179.
[32] Sridhar R, Yang D. Explaining Toxic Text via Knowledge Enhanced Text Generation. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, United States: Association for Computational Linguistics; 2022. p. 811-26. Available from: https://aclanthology.org/2022.naacl-main.59.
[33] Haimson OL, Delmonaco D, Nie P, Wegner A. Disproportionate Removals and Differing Content Moderation Experiences for Conservative, Transgender, and Black Social Media Users: Marginalization and Moderation Gray Areas. Proc ACM Hum-Comput Interact. 2021 oct;5(CSCW2). Available from: https://doi-org.libezproxy.open.ac.uk/10.1145/3479610.
[34] Thiago DO, Marcelo AD, Gomes A. Fighting hate speech, silencing drag queens? artificial intelligence in content moderation and risks to lgbtq voices online. Sexuality & culture. 2021;25(2):700-32.
[35] Kronk CA, Dexheimer JW. Development of the Gender, Sex, and Sexual Orientation ontology: Evaluation and workflow. Journal of the American Medical Informatics Association. 2020 06;27(7):1110-5. Available from: https://doi.org/10.1093/jamia/ocaa061.
[36] Breit A, Waltersdorfer L, Ekaputra FJ, Sabou M, Ekelhart A, Iana A, et al. Combining Machine Learning and Semantic Web: A Systematic Mapping Study. ACM Comput Surv. 2023 mar. Available from: https://doi.org/10.1145/3586163.
[37] Vrandečić D, Krötzsch M. Wikidata: A Free Collaborative Knowledgebase. Commun ACM. 2014 sep;57(10):78–85. Available from: https://doi.org/10.1145/2629489.
[38] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference. ISWC'07/ASWC'07. Berlin, Heidelberg: Springer-Verlag; 2007. p. 722–735.
[39] Suchanek FM, Kasneci G, Weikum G. Yago: A Core of Semantic Knowledge. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07. New York, NY, USA: Association for Computing Machinery; 2007. p. 697–706. Available from: https://doi.org/10.1145/1242572.1242667.
[40] d'Avila Garcez A, Lamb LC. Neurosymbolic AI: The 3rd Wave. arXiv e-prints. 2020 Dec:arXiv:2012.05876.
[41] Hamilton K, Nayak A, Božić B, Longo L. Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. Semantic Web. 2022 nov:1-42. Available from: https://doi.org/10.3233%2Fsw-223228.
[42] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.
[43] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[44] Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, John RS, et al. Universal sentence encoder. arXiv preprint arXiv:180311175. 2018.
[45] Borkan D, Dixon L, Sorensen J, Thain N, Vasserman L. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. CoRR. 2019;abs/1903.04561. Available from: http://arxiv.org/abs/1903.04561.
[46] Maronikolakis A, Wisiorek A, Nann L, Jabbar H, Udupa S, Schuetze H. Listening to Affected Communities to Define Extreme Speech: Dataset and Experiments. In: Findings of the Association for Computational Linguistics: ACL 2022. Dublin, Ireland: Association for Computational Linguistics; 2022. p. 1089-104. Available from: https://aclanthology.org/2022.findings-acl.87.
[47] Mathew B, Saha P, Yimam SM, Biemann C, Goyal P, Mukherjee A. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. Proceedings of the AAAI Conference on Artificial Intelligence. 2021 May;35(17):14867-75. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/17745.
[48] Braun V, Clarke V. Using thematic analysis in psychology. Qualitative research in psychology. 2006;3(2):77-101.
[49] Satopaa V, Albrecht J, Irwin D, Raghavan B. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops. IEEE; 2011. p. 166-71.
[50] Janowicz K, Yan B, Regalia B, Zhu R, Mai G. Debiasing Knowledge Graphs: Why Female Presidents

are not like Female Popes. In: ISWC (P&D/Industry/BlueSky); 2018. .

[51] Röttger P, Vidgen B, Nguyen D, Waseem Z, Margetts H, Pierrehumbert J. HateCheck: Functional Tests for Hate Speech Detection Models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics; 2021. p. 41-58. Available from: https://aclanthology.org/2021.acl-long.4.

[52] Calabrese A, Bevilacqua M, Ross B, Tripodi R, Navigli R. AAA: Fair Evaluation for Abuse Detection Systems Wanted. WebSci '21. New York, NY, USA: Association for Computing Machinery; 2021. p. 243–252. Available from: https://doi-org.libezproxy.open.ac.uk/10.1145/3447535.3462484.

[53] Leonardelli E, Menini S, Aprosio AP, Guerini M, Tonelli S. Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators' Disagreement. arXiv preprint arXiv:210913563. 2021.

[54] Keyes O. The Misgendering Machines: Trans/HCI Implications of Automatic Gender Recognition. Proc ACM Hum-Comput Interact. 2018 nov;2(CSCW). Available from: https://doi-org.libezproxy.open.ac.uk/10.1145/3274357.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230003

Using Pre-Trained Language Models for Abstractive DBPEDIA Summarization: A Comparative Study

Hamada M. ZAHERA♣ [a,1], Fedor VITIUGIN♣ [b], Mohamed Ahmed SHERIF [a], Carlos CASTILLO [b,c] and Axel-Cyrille NGONGA NGOMO [a]

[a] *Data Science Group, Department of Computer Science, Paderborn University*
[b] *Web Science and Social Computing Research group, Universitat Pompeu Fabra, Spain*
[c] *ICREA, Catalan Institution for Research and Advanced Studies, Spain*
ORCiD ID: Hamada M. Zahera https://orcid.org/0000-0003-0215-1278, Fedor Vitiugin https://orcid.org/0000-0003-4350-1828, Mohamed Ahmed Sherif https://orcid.org/0000-0002-9927-2203, Carlos Castillo https://orcid.org/0000-0003-4544-0416, Axel-Cyrille Ngonga Ngomo https://orcid.org/0000-0001-7112-3516

Abstract.

Purpose: This study addresses the limitations of current short abstracts of DBPEDIA entities, which often lack a comprehensive overview due to their creating method (i.e., selecting the first two-three sentences from the full DBPEDIA abstracts).

Methodology: We leverage pre-trained language models to generate abstractive summaries of DBPEDIA abstracts in six languages (English, French, German, Italian, Spanish, and Dutch). We performed several experiments to assess the quality of generated summaries by language models. In particular, we evaluated the generated summaries using human judgments and automated metrics (Self-ROUGE and BERTScore). Additionally, we studied the correlation between human judgments and automated metrics in evaluating the generated summaries under different aspects: informativeness, coherence, conciseness, and fluency.

Findings: Pre-trained language models generate summaries more concise and informative than existing short abstracts. Specifically, BART-based models effectively overcome the limitations of DBPEDIA short abstracts, especially for longer ones. Moreover, we show that BERTScore and ROUGE-1 are reliable metrics for assessing the informativeness and coherence of the generated summaries with respect to the full DBPEDIA abstracts. We also find a negative correlation between conciseness and human ratings. Furthermore, fluency evaluation remains challenging without human judgment.

Value: This study has significant implications for various applications in machine learning and natural language processing that rely on DBPEDIA resources. By providing succinct and comprehensive summaries, our approach enhances the quality of DBPEDIA abstracts and contributes to the semantic web community.

Keywords., Abstractive Summarization, Large Language Models, Knowledge Graphs.

♣Equal Contribution
[1]Corresponding Author: Hamada M. Zahera; E-mail: hamada.zahera@uni-paderborn.de

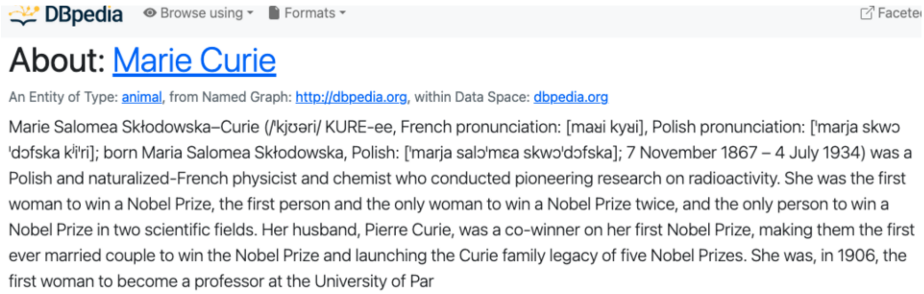


Figure 1. An example of shortened abstract of "Marie Curie" entity in DBPEDIA.

1. Introduction

DBPEDIA is one of the most popular knowledge graphs in the Linked Open Data cloud (LOD) [1]. DBPEDIA has been widely used as a significant resource for accessing and linking knowledge on the web, particularly in the context of the semantic web and linked data. Entity abstracts (`dbo:abstract`) are an essential component of DBPEDIA, as they provide a concise summary of the Wikipedia page for each entity. Moreover, there are two types of DBPEDIA abstracts: 1) *Full abstracts*, which are extracted from first paragraphs of the corresponding WIKIPEDIA article for each entity. 2) *Short abstracts*[2] are automatically created by selecting the first few sentences (i.e., two-three sentences) from the full abstracts [2]. Short abstracts are used to provide users with a comprehensive overview of the most significant information about entities. For example, *Google* employs short abstracts of search concepts in the knowledge panel to offer users a concise summary of the searched entities [3]. However, the method of creating these short abstracts omits other relevant information in the remaining portion of the full abstract. Figure 1 shows an example of the shortened abstract of *"Marie Curie"* entity that is created by truncating[3] the first sentences from its full abstract[4]. This shortened abstract ignores other essential information such as *"The cause of her death was given as aplastic pernicious anaemia, a condition she developed after years of exposure to radiation through her work"*, which is relevant for understanding *Marie Curie's* life and achievements. It is important to note that some short abstracts of DBPEDIA are unavailable in specific languages.

To address these challenges, we leverage pre-trained language models (LLMs) to generate abstractive summaries of DBPEDIA entities. Recently advances in pre-trained language models have led to impressive performance in text summarization tasks, achieving state-of-the-art performance on various benchmark datasets [4–7]. Inspired by this success, we employ two state-of-the-art LLMs in our comparative study: i) BART (short for *Bidirectional Auto-Regressive transformers*) model, which can generate more accurate and coherent summaries by considering the context of a text in both directions (left-to-right and right-to-left) [8], ii) T5 (short for *Text-To-Text Transfer Transformer*) model is based on a transformer architecture with a self-attention mechanism that uses a text-to-text approach, i.e., the T5 model is trained to generate an output text based on an

[2] https://databus.dbpedia.org/dbpedia/text/short-abstracts/
[3] full text of last sentence is "*the first woman to become a professor at the University of Paris*"
[4] https://en.wikipedia.org/wiki/Marie_Curie

input. This allows the T5 model to be used across various tasks (e.g., text summarization, question answering, machine translation). To ensure the accessibility and affordability of our summarization approach, we chose these open-source models (BART and T5) over commercial models (e.g., GPT-3, GPT-4) which require API subscriptions (e.g., OpenAI API) or large computational resources. Moreover, open-source models offer high adaptability and can be readily fine-tuned on domain-specific datasets with minimal effort. Furthermore, previous studies have demonstrated that both BART and T5 can generate summaries of comparable quality to those produced by smaller GPT-3 models [9–11].

We performed several experiments to identify the most suitable pre-trained LLM for generating abstractive summaries of DBPEDIA abstracts in six languages. We used DBPEDIA abstracts in *English, German, French, Italian, Spanish*, and *Dutch* as our evaluation dataset and produced summaries using various LLMs. We then evaluated the quality of the LLMs-generated summaries against the existing shortened abstracts using both human judgments and automated metrics. Furthermore, we investigated the correlation between the automated metrics and human assessments of the summaries' quality. Our evaluation results indicated that LLMs are effective tools for creating informative summaries for DBPEDIA abstracts. However, the choice of LLMs should be adapted to the specific language. We summarize the main contributions of our study as follows:

- To the best of our knowledge, this is the first study to leverage LLMs to generate abstractive summaries of DBPEDIA abstracts compared to the existing method that automatically selects the first few sentences from the full abstracts.
- We compared the performance of different LLMs for generating abstractive summaries in six languages (English, German, French, Italian, Spanish, and Dutch) using human and automated evaluation metrics
- We analyzed the correlation between the automated metrics (BERTScore and self-ROUGE) and the human judgments of the quality of generated summaries.
- We provide a resource of abstractive summaries of all DBPEDIA abstracts (v2022) in English and German.[5]

2. Related Works

LLMs for abstractive summarization. Recent years have witnessed a growing interest in summarizing descriptions of real-world entities in knowledge graphs [12, 13]. This task, known as text summarization, requires selecting the most essential and salient concepts, entities, and relationships from the knowledge graph, and generating a brief and coherent summary of them. Text summarization can generally be divided into two categories: i) *extractive summarization* [14], which involves selecting the most salient and informative sentences from a document to create a summary, and ii) *abstractive summarization* [15], which involves generating a new summary that conveys the main ideas of the original document, potentially using new phrases and sentences that were not present in the original text. Our study focuses on the latter for generating abstractive summaries of DBPEDIA abstracts.

Abstractive summarization is a text-generation process that aims to produce summaries that are fluent and coherent, as well as informative and concise. Previous works

[5]https://zenodo.org/record/7600894

have employed deep neural networks and language generation techniques to achieve this goal, often using a sequence-to-sequence (Seq2Seq) architecture with an attention mechanism or transformers. These methods can generate summaries that are more expressive and natural than extractive summaries, which simply select sentences from the original document. For example, See et al. [16] proposed the pointer-generator network, which combines the ability to generate new words with the ability to copy words from the input text. This hybrid approach allows for the generation of more fluent and accurate summaries as demonstrated by the evaluation results on the *CNN/Daily Mail* dataset, where it outperformed several baselines. Another example is the fine-tuning of pre-trained language models on large-scale summarization datasets, which can lead to substantial improvements in abstractive summarization and generate higher-quality summaries [17]. Pre-trained language models such as T5, BART, and GPT-2 have also achieved outstanding performance in generating high-quality summaries in terms of relevance, fluency, and semantic accuracy [18] Motivated by this success, we propose our approach for employing pre-trained language models to produce abstractive summaries of DBPEDIA abstracts. To the best of our knowledge, this is the first study to apply language models to this task. The existing method for creating summaries of DBPEDIA abstracts (i.e., short abstracts) simply selects the first few sentences from each entity's description.

Evaluating LLM-generated summaries. Evaluating the quality of generated summaries by large language models is a challenging task [19]. One approach is to use manual evaluation, where human experts are asked to grade the summaries based on their understanding and perception of the content [20]. For example, Iskender et al. [21] compared crowdsourcing ratings with expert ratings and automatic metrics such as ROUGE, BLEU, or BERTScore on a German summarization dataset. They found that crowdsourcing can be used as a direct substitute for experts when measuring structure and coherence, but should be considered carefully when judging overall quality, grammaticality, clarity, and summary informativeness. On the other hand, researchers have proposed self-evaluation methods such as BERTScore [22] and Self-ROUGE [23, 24] that compare the quality of generated summaries with respect to the original text. Specifically, the BERTScore metric measures the semantic similarity between a generated summary and its corresponding original text using cosine distance between their contextualized BERT embeddings [22]. For instance, Koroteev [25] demonstrated the use of semantic text-similarity metrics for evaluating the quality of abstractive summaries in Russian. The author argues that semantic text-similarity metrics are a valuable tool for a variety of natural language processing (NLP) tasks, such as machine translation, information retrieval, and text summarization. Due to the lack of gold-standard summaries for DBPEDIA abstracts, we follow the evaluation methods used by previous works [22, 23, 26] that employed BERTScore and Self-ROUGE as well as crowdsourcing evaluation to assess the quality of the generated summaries in our experiments. We provide more details about these evaluation metrics in Section 4.3.

3. Approach

This section explains the preprocessing steps for the input data (DBPEDIA abstracts), followed by the description of the pre-trained models used in our study. Figure 2 depicts the complete pipeline of our approach, which generates abstractive summaries for DBPEDIA abstracts.

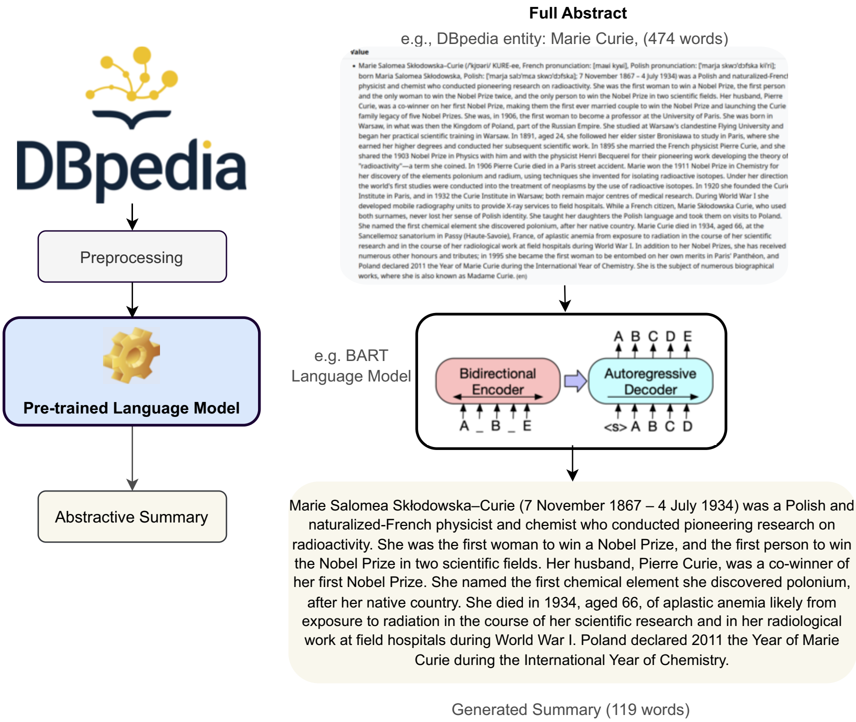


Figure 2. The pipeline of abstractive summarization of DBPEDIA using language models.

3.1. *Preprocessing*

We note that advanced language models such as BART and T5 are pre-trained on large-scale text corpora and can handle variations in capitalization, stopwords, and word forms [27]. Thus, we do not need to preprocess the text with lowercase, stopword removal, and stemming or lemmatization before applying these models for text summarization. However, we need to format the input text according to the specific requirements of the language models [8, 28]

- *Tokenization:* Tokenization is the process of breaking down text into smaller units, called tokens, that can be characters, subwords, or words. Language models require input text to be tokenized using their own tokenizers, which handle punctuation and special characters appropriately as well as maintain compatibility with the model's preprocessing requirements.
- *Truncating and Padding*: To ensure a uniform length of input sequences for language models, input text that is longer or shorter than a predefined maximum length needs to be padded or truncated. The padding process involves appending special tokens, such as $\langle pad \rangle$, to the end of shorter sequences, while truncation requires removing excess tokens from longer sequences.
- *Formatting*: Language models require specific input formatting to distinguish between different tasks. For a text summarization task, a task prompt (e.g., "summarize") should be used to indicate the desired output.

- *Handling Special Tokens*: Language models use a set of unique tokens, like $\langle eos \rangle$, $\langle bos \rangle$, $\langle unk \rangle$, and $\langle pad \rangle$ to indicate the start/end of a sentence, unknown words, and padding, respectively. It is essential to incorporate these tokens into the input text during preprocessing to ensure proper functioning.
- *Post-processing*: After generating summaries, it may be necessary to conduct post-processing steps to improve the readability and coherence of the output. These steps may include removing redundant or irrelevant tokens, reassembling the sentence structure, and applying appropriate capitalization and punctuation.

3.2. Pre-trained Language Models for Abstractive DBPEDIA *Summarization*

With the advent of pre-trained language models, the field of NLP has been revolutionized, resulting in significant improvements in various tasks, including abstractive summarization [29]. BART [8] and T5 [28] are among the state-of-the-art models for abstractive text summarization. We summarize each mode as follows:

- BART model is a denoising autoencoder that employs a bidirectional encoder and a left-to-right decoder. This model is pre-trained on a large-scale corpus by reconstructing the original text after being corrupted by various noise functions, such as token masking and sentence permutation. This pre-training strategy enables BART to learn a rich latent space representation of the input text, which is useful for generating coherent and contextually relevant summaries. Moreover, BART has exhibited strong performance in abstractive summarization tasks, outperforming previous state-of-the-art models on the benchmark summarization CNN/Daily Mail and XSum datasets [30].
- T5 model is another powerful language model based on the transformer architecture. It is designed with a unified text-to-text framework, which allows fine-tuning on different NLP tasks by simply converting them into text-to-text problems. Additionally, T5's pre-training objective, which involves reconstructing corrupted input text, enables it to learn rich representations that can be leveraged for generating abstractive summaries [31].

4. Evaluation

We conducted our experiments to answer the following research questions:

Q_1: Which LLM is suitable for generating summaries of DBPEDIA abstracts in which language, based on human evaluation and automated similarity metrics?

Q_2: What is the correlation between human ratings and automated metrics in evaluating the informativeness, coherence, conciseness, and fluency of the generated summaries?

4.1. Evaluation Dataset

Our goal is to evaluate the performance of pre-trained large language models in summarizing DBPEDIA abstracts. For this purpose, we created a dataset of 600 DBPEDIA abstracts in six languages (English, German, French, Spanish, Dutch, and Italian), with

100 abstracts randomly selected for each language. We selected the target languages based on the availability of *Short abstracts* dataset except for Japanese due to its special tokenization process. Table 1 provides a statistical overview including *the number of abstracts* in each language and *the average number of sentences*.

4.2. Models

We employed four different models in our study: three variants of the BART model ($\text{BART}_{\text{large-50}}$, $\text{BART}_{\text{large-CNN}}$, and $\text{BART}_{\text{weak-sup}}$) and the pre-trained T5_{large} model. We provide a brief description of each baseline as follows:

- $\text{BART}_{\text{large-50}}$ is a multilingual model with 139M parameters, 12 layers, and a hidden size of 768 and supported 50 languages [32].
- $\text{BART}_{\text{large-CNN}}$ is a large-scale variant of BART model with 400M parameters, 12 encoder, and decoder layers. Furthermore, the model was fine-tuned on a collection of news articles and their golden-standard summaries from *CNN/DailyMail* dataset [33].
- $\text{BART}_{\text{weak-sup}}$ is a weakly-supervised BART model [34], which is fine-tuned via incorporating rich external knowledge from CONCEPTNET [35].
- T5_{LARGE} [28] is a pre-trained text-to-text transformer model that can generate text for different NLP tasks. It has 770M parameters and is trained on a large corpus of web texts using a masked language modelling objective.

4.3. Evaluation Metrics

Automated Evaluation. To evaluate the quality of LLMs-generated summaries with respect to the full DBPEDIA abstracts, we employ the following metrics:

- *Self-ROUGE* is a self-evaluation metric that measures the similarity between the generated summaries and the original text by computing their n-gram overlaps [36]. Due to the lack of gold-standard summaries for DBPEDIA abstracts, we employ Self-ROUGE to extract n-grams tokens from both the generated summaries and the full DBPEDIA abstracts and calculate the Precision, Recall, and F_1 scores based on the n-grams overlaps (ROUGE metric). Following previous works [23, 26, 37], we selected the top-3 sentences with the highest ROUGE scores (i.e., the ROUGE scores of each sentence when using the rest of the sentences as the reference summary) as the reference text (*silver-standard summaries*) in a greedy manner.
- *BERTScore* [22] measures the similarity between the generated text and the reference text using contextualized embeddings from the pre-trained BERT model. In our study, we employ the full DBPEDIA abstract as a reference text, since there are no golden summaries for the DBPEDIA abstracts. Moreover, we obtain the embedding vector for each token in LLMs-generated summaries ($x = x_1, x_2, \cdots, x_{|x|}$) and DBPEDIA full abstracts ($y = y_1, y_2, \cdots, y_{|y|}$) from the pre-trained BERT model. Each token $x_i \in x$ is aligned to the most similar token in $y_i \in y$ and vice-versa. To achieve this, we compute the pairwise cosine similarity between each token in the generated summary ($x_i \in x$) and each in its corresponding in the full abstract ($y_i \in y$). The cosine similarity is defined as $cos(x_i, y_j) = \frac{x_i^T \cdot y_j}{||x|| \cdot ||y||}$. In LLMs, the embeddings are typically normalized to a unit vector, i.e. $||x||$ and $||y||$ are 1, therefore this

Table 1. The statistics of evaluation dataset

	English	Spanish	German	French	Italian	Dutch
Number of abstracts	100	100	100	100	100	100
Average number of sentences	6.5	4.98	5.6	3.4	3.17	6.3

computation is simplified to $x_i^T \cdot y_j$. Furthermore, Precision (P), Recall (R), and F_1 scores are computed based on BERTscores as follows:

$$P_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{y_j \in y} x_i^T \cdot y_j \tag{1}$$

$$R_{\text{BERT}} = \frac{1}{|y|} \sum_{y_j \in y} \max_{x_i \in x} x_i^T \cdot y_j \tag{2}$$

$$F_{1_{\text{BERT}}} = 2 \times \frac{R_{\text{BERT}} \cdot P_{\text{BERT}}}{R_{\text{BERT}} + P_{\text{BERT}}} \tag{3}$$

Human Evaluation. In the absence of reference summaries, crowdsourcing services have become an effective alternative to easily and quickly recruit users (i.e., crowdworkers) in performing manual evaluations of DBPEDIA abstractive summarization. We used the *SurgeHQ*[6] crowdsourcing platform to conduct our experiments, as illustrated in Figure 3. We bounded the evaluation of generated summaries to crowdworkers who are fluent in the target languages. The evaluation procedure contained two main tasks: i) crowdworkers were instructed to select the most appropriate summary that best summarized the full DBPEDIA abstracts. In particular, they compared the summaries LLMs-generated summaries to the shortened DBPEDIA abstracts and ii) they rated each summary, including the shortened ones, using a 4-point *Likert scale*, according to the following criteria:

- *Informativeness* measures how well a generated summary captures the essential information in the source text. A summary is informative if it accurately represents the main ideas and critical points of the original content.
- *Coherence* relates to the logical flow and organization of the summary, ensuring that the ideas and concepts are clearly presented and connected. A summary is coherent if it is easily comprehensible and maintains a well-structured narrative that conveys the main points without confusion.
- *Conciseness* assesses how effectively a summary expresses the essential information from the source text in a clear and succinct manner, without unnecessary repetition or wordiness. A summary is concise if it conveys the key points using the minimum number of words possible, making it an efficient representation of the original content.

[6]https://www.surgehq.ai/

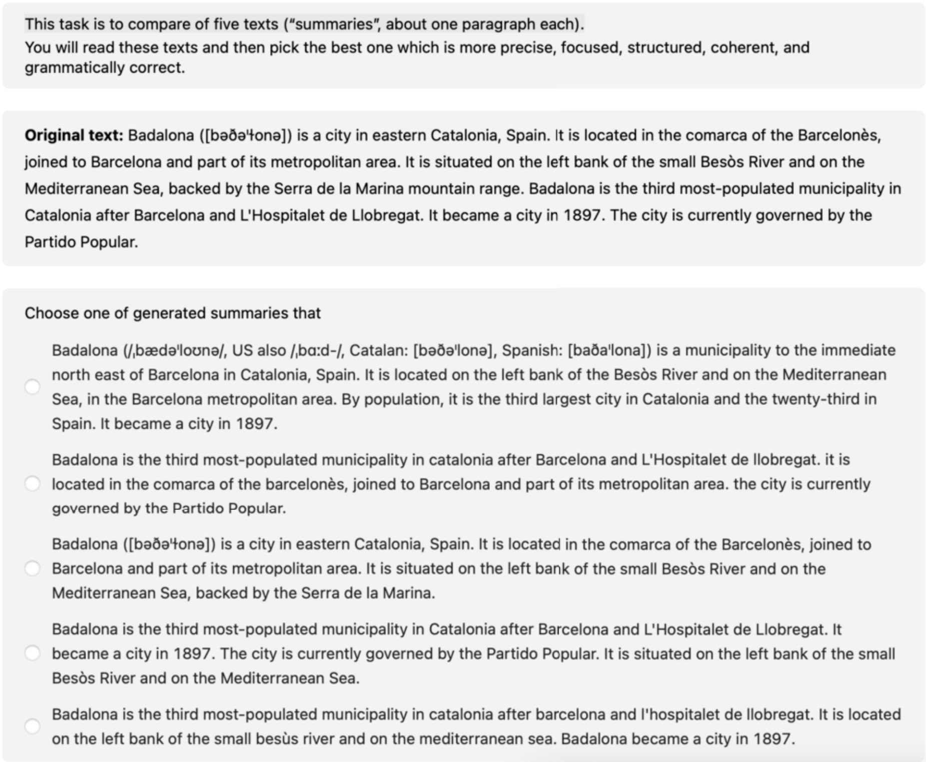


Figure 3. An example of a crowdsourcing task for Barcelona city. Human annotators were asked to select the most informative summary.

- *Fluency* evaluates the naturalness and readability of the generated summary. A summary is fluent if it has smooth and effortless expression, with proper grammar, syntax, and punctuation.

To ensure the reliability of our evaluation, we asked three crowdworkers to assess each summary using these criteria. We then computed the average scores for all the generated summaries.

5. Results

To answer $\mathbf{Q}_1$, we adopted various evaluation metrics to assess the quality of LLMs-generated abstracts. Automated summarization techniques such as Self-ROUGE and BERTScore were used to quantify the models' performance. A human evaluation was also conducted to assess the quality of the summaries generated under different aspects.

5.1. Automated evaluation of LLMs-generated summaries

Self-ROUGE evaluation. Table 2 presents the evaluation results of ROUGE scores for all LLMs-generated summaries and short abstracts. We observe that $\text{BART}_{\text{large-50}}$

This task is to evaluate the quality of generated text ("summary", about one paragraph each). You will read the original text and then evaluate generated text in four dimensions: informativeness, coherence, conciseness, and fluency.

Source Text: Watchmen is a comic-book limited series written by Alan Moore, artist Dave Gibbons, and colorist John Higgins published by DC Comics in 1986 and 1987, and collected in 1987. Watchmen originated from a story proposal Moore submitted to DC featuring superhero characters that the company had acquired from Charlton Comics. As Moore's proposed story would have left many of the characters unusable for future stories, managing editor Dick Giordano convinced Moore to create original characters instead. Moore used the story as a means to reflect contemporary anxieties and to deconstruct and parody the superhero concept. Watchmen depicts an alternate history where superheroes emerged in the 1940s and 1960s, helping the United States to win the Vietnam War. In 1985, the country is edging toward nuclear war with the Soviet Union, freelance costumed vigilantes have been outlawed and most former superheroes are in retirement or working for the government. The story focuses on the personal development and moral struggles of the protagonists as an investigation into the murder of a government sponsored superhero pulls them out of retirement. Creatively, the focus of Watchmen is on its structure. Gibbons used a nine-panel grid layout throughout the series and added recurring symbols such as a blood-stained smiley face. All but the last issue feature supplemental fictional documents that add to the series' backstory, and the narrative is intertwined with that of another story, a fictional pirate comic titled Tales of the Black Freighter, which one of the characters reads. Structured as a nonlinear narrative, the story skips through space, time and plot. In the same manner, entire scenes and dialogue have parallels with others through synchronicity, coincidence and repeated imagery. A commercial success, Watchmen has received critical acclaim both in the comics and mainstream press, and is considered by several critics and reviewers as one of the most significant works of 20th century literature. After a number of attempts to adapt the series into a feature film, director Zack Snyder's Watchmen was released in 2009. A video game series, Watchmen: The End is Nigh, was released in the same year to coincide with the film's release. In 2012, DC Comics began publishing Before Watchmen, a comic book series acting as a prequel to the original Watchmen series, without Moore and Gibbons' involvement. Watchmen was recognized in Time's List of the 100 Best Novels as one of the best English language novels published since 1923, and placed #91 on The Comics Journal's list of the top 100 comics of the 20th century.

Summary: Watchmen is a comic-book limited series written by Alan Moore, artist Dave Gibbons, and colorist John Higgins published by DC Comics in 1986 and 1987, and collected in 1987. Watchmen originated from a story proposal Moore submitted to DC featuring superhero characters that the company had acquired from Charlton Comics. As Moore's proposed story would have left many of the characters unusable for future stories, managing editor Dick Giordano convinced Moore to create original characters instead. Moore used the story as a means to reflect contemporary anxieties and to deconstruct and parody the superhero concept. Watchmen depicts an alternate history where superheroes emerged in the 1940s and 1960s, helping the United States to win the Vietnam War. In 1985, the country is edging toward nuclear war with the Soviet Union, freelance costumed vigilantes have been

Evaluate the **informativeness** of the summary compared to the source text, where:

- **1** — while the summary lose the crucial information from the source text at all;
- **4** — the ideal summary which captures all important information.

○ 1

○ 2

○ 3

○ 4

Figure 4. An example of a crowdsourcing task for evaluating the informativeness of generated summary.

generates high-quality summaries for most languages, except for Dutch where the shortened abstracts outperform the LLMs-generated summaries. Using a common threshold of p-value $= 0.05$ for significance testing[7], the results indicate a significant difference in score values (ROUGE-1 p-value ≤ 0.05; ROUGE-2 p-value ≤ 0.06) of $\text{BART}_{\text{large-50}}$ and short abstract.

BERTScore evaluation. Table 3 presents the evaluation results of LLMs-generated summaries and short abstracts using F_{BERT} as computed in Equation (3). Among all models, $\text{BART}_{\text{large-50}}$ achieves the best performance for most languages, indicating its effectiveness in generating high-quality summaries of DBpedia abstracts in multiple languages. However, for English, the quality of short abstracts is better by $+3.39\%$.

[7]We tested if BARTlarge-50 has higher score values than short abstract using hypotheses (H0: No difference in score values) (H1: BARTlarge-50 has higher score values)

Table 2. Self-ROUGE evaluation results: *ROUGE-1* (R1), and *ROUGE-2* (R2)

	English		Spanish		German		French		Italian		Dutch	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
Short-abstracts	0.58	**0.62**	0.51	0.45	0.68	0.60	0.70	0.61	0.64	0.54	**0.66**	**0.57**
T5	0.52	0.40	0.57	0.45	0.57	0.45	0.65	0.55	0.67	0.59	0.59	0.49
$\text{BART}_{\text{large-50}}$	**0.61**	0.53	**0.72**	**0.66**	**0.78**	**0.74**	**0.83**	**0.79**	**0.83**	**0.80**	0.63	0.55
$\text{BART}_{\text{large-CNN}}$	**0.61**	0.53	0.58	0.47	0.61	0.51	0.64	0.55	0.67	0.58	0.58	0.45
$\text{BART}_{\text{weak-sup}}$	0.49	0.34	0.30	0.16	0.24	0.10	0.25	0.12	0.26	0.13	0.34	0.20

Table 3. BERTScore (F1) evaluation results

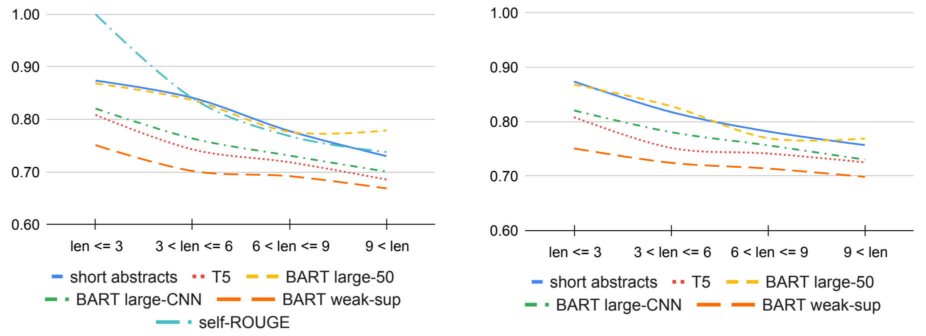
	English	Spanish	German	French	Italian	Dutch
Short-abstract	**0.87**	**0.75**	0.86	0.72	0.86	0.81
T5	0.83	0.68	0.75	0.70	0.81	0.70
$\text{BART}_{\text{large-50}}$	0.84	**0.75**	**0.89**	**0.84**	**0.88**	**0.84**
$\text{BART}_{\text{large-CNN}}$	0.84	0.72	0.76	0.74	0.80	0.73
$\text{BART}_{\text{weak-sup}}$	0.83	0.66	0.67	0.66	0.72	0.68

Therefore, we performed an in-depth analysis based on the number of sentences in each abstract. We grouped the DBPEDIA abstracts used in our experiments into four categories: i) *up to 3 sentences*, same as to the short abstracts consisting of the first three sentences of the original articles (40% of original abstracts), ii) *from 4 to 6 sentences*, which is twice the length of short abstracts (25% of original abstracts), iii) *from 7 to 9 sentences*, which adds three more sentences to the previous group (15% of original abstracts), and iv) *more than 9 sentences*, which forms the final bin (19% of original abstracts). As shown in Figure 5a, $\text{BART}_{\text{large-50}}$ model achieves comparable BERTScores to short abstracts for DBPEDIA abstracts up to 9 sentences and surpasses them for longer abstracts. For other models, we observed that BERTScore decrease as original texts become longer. As shown in Figure 5b BERTScore for short abstracts and summaries generated by $\text{BART}_{\text{large-50}}$ compared to Self-ROUGE summaries are similar. These plots indicate that $\text{BART}_{\text{large-50}}$ summaries achieve higher BERTScore scores than short abstracts, especially for longer texts. Overall, our results conclude that $\text{BART}_{\text{large-50}}$ is an effective resource for generating high-quality summaries of DBPEDIA abstracts depending on their lengths and can help guide future research studies.

5.2. *Human Evaluation of LLM-generated summaries of* DBPEDIA *abstracts*

We conducted two crowdsourcing experiments to evaluate the generated summaries in six languages: English, Spanish, German, French, Italian, and Dutch.

In the *first experiment*, we presented 100 abstracts per language to native speakers and asked them to choose the most comprehensive summary between a short abstract, or LLMs-generated summaries by $\text{BART}_{\text{large-CNN}}$, $\text{BART}_{\text{large-50}}$, or T5. For each abstract, we used a majority vote of three annotators to select the best summary. Table 4 shows the percentage of summaries chosen by the annotators for each language and model. We observe that 36% of the human annotators preferred the generated summaries by $\text{BART}_{\text{large-CNN}}$, 45% preferred the summaries generated by the T5 model in German, and



(a) BERTScore similarity with original DBPEDIA abstracts **(b)** BERTScore similarity with self-ROUGE summaries

Figure 5. BERTScore for abstracts with different sentence lengths in 6 languages.

Table 4. Human evaluation of the LLM-generated summaries in 6 languages. The average rate of annotators' agreement = 0.71

	English	Spanish	German	French	Italian	Dutch
Short-abstracts	28%	**48%**	32%	**36%**	**46%**	35%
T5	4%	2%	**45%**	25%	12%	9%
BART$_{\text{large-50}}$	22%	42%	6%	24%	34%	**36%**
BART$_{\text{large-CNN}}$	**36%**	8%	6%	15%	7%	16%
BART$_{\text{weak-sup}}$	9%	0%	11%	0%	1%	4%

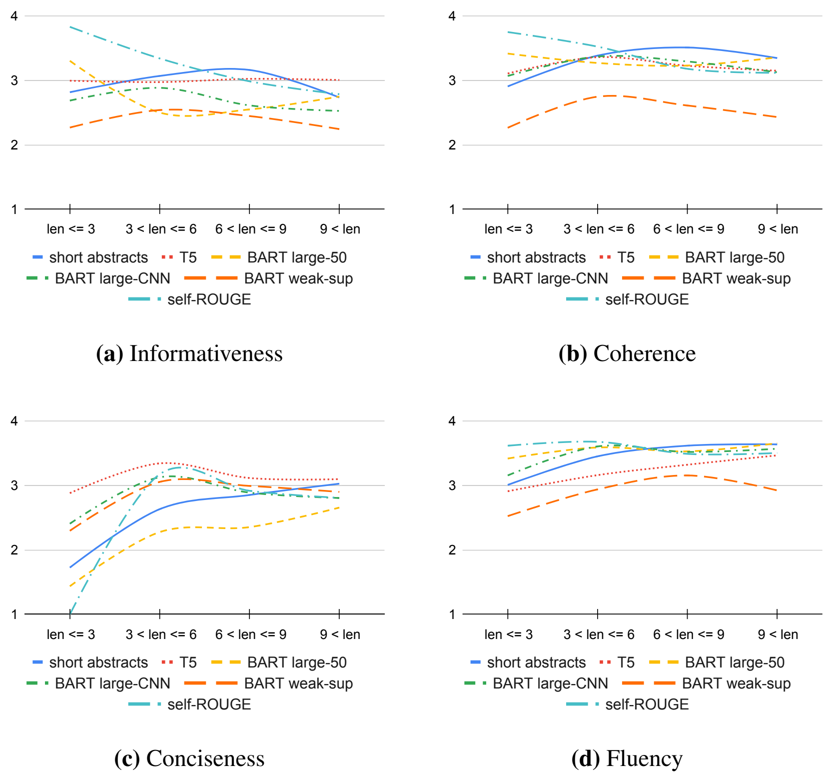
36% selected the BART$_{\text{large-50}}$-generated summaries in Dutch. For Spanish, French, and Italian languages, the annotators selected short abstracts instead. These results suggest that the length of DBPEDIA abstracts influences human preferences. For shorter abstracts (less than five sentences), human annotators preferred short abstracts. For longer abstracts (more than five sentences), they selected the LLMs-generated summaries. This implies that short abstracts are informative enough in the case of full DBPEDIA abstracts with short content and do not need further summarization. In contrast, longer DBPEDIA abstracts can be summarized efficiently using pre-trained large language models.

In the *second experiment*, we performed another crowdsourcing evaluation to assess the quality of the generated summaries and short abstracts based on four criteria: *informativeness, coherence, conciseness*, and *fluency*. We used a 4-point scale, where 1 is the lowest and 4 is the highest rating. Each summary was compared with the original DBPEDIA abstract by three crowdworkers, following the same procedure as in the first experiment. The evaluation results in Table 5 demonstrate that the T5 model outperforms the other models in terms of informativeness and conciseness, whereas the BART$_{\text{large-50}}$ model performs better in terms of coherence and fluency.

Finally, we performed an in-depth analysis of the generated summaries and short abstracts based on their length, in the same manner, in Section 5.1. We used the same categorization of DBPEDIA abstracts based on the number of sentences Figure 6. We observe that T5 and BART$_{\text{large-50}}$ produced more informative and coherent summaries than short abstracts for DBPEDIA summaries with more than 9 sentences. Moreover,

Table 5. Human evaluation of the quality of the LLM-generated summaries (average scores of the English, German, and Dutch languages). The average rate of annotators' agreement = 0.69

Model	**Informativeness**	**Coherence**	**Conciseness**	**Fluency**
Short-abstract	2.94	3.28	2.55	3.42
T5	**2.99**	3.21	**3.12**	3.21
$\text{BART}_{\text{large-50}}$	2.77	**3.32**	2.18	**3.55**
$\text{BART}_{\text{large-CNN}}$	2.68	3.21	2.81	3.46
$\text{BART}_{\text{weak-sup}}$	2.37	2.51	2.81	2.88



(a) Informativeness

(b) Coherence

(c) Conciseness

(d) Fluency

Figure 6. Human evaluation of generated summaries in different criteria for DBPEDIA abstracts (English) with different sentence lengths.

$\text{BART}_{\text{large-50}}$ model created more fluent summaries than short abstracts for most categories. Interestingly, T5 model produced more concise summaries than short abstracts, regardless of their length. In summary, the human evaluation indicates that both models $\text{BART}_{\text{large-50}}$ and T5 can produce summaries of equivalent quality. In the automated evaluation using Self-ROUGE and BERTScore metrics, the $\text{BART}_{\text{large-50}}$ model generated better summaries than other models and short abstracts.

5.3. Automated and Human evaluation results correlation

To answer $\mathbf{Q}_2$, we measured the correlation between the scores of automatic metrics and human judgments using two non-parametric rank correlation coefficients: *Spearman's rank* and *Kendall's rank*. Specifically, *Spearman's rank correlation coefficient*, denoted by Spearman's ρ, assesses the linear association between two variables based on their ranks [38]. Similarly, *Kendall rank correlation coefficient*, denoted by Kendall's τ, evaluates the degree of agreement between two ranked variables [39]. We computed both coefficients for the single document task [40] and plotted them in Figure 7. These measures do not require any assumptions about the distribution of the variables or their joint distribution. Our correlation analysis indicates that BERTScore has the strongest relationship with human ratings of *informativeness*, with Spearman and Kendall coefficients of $\rho \leq 0.61$ and $\tau \leq 0.49$, respectively. Furthermore, ROUGE-1 has the highest correlation with human assessment of *coherence*, with Spearman and Kendall coefficients of $\rho \leq 0.31$ and $\tau \leq 0.25$, respectively. We also observe that *conciseness* has a negative correlation with human evaluation in all cases, with Spearman and Kendall coefficients of $\rho \geq -0.62$ and $\tau \geq -0.52$, while *fluency* has a negligible correlation with values close to 0. Therefore, BERTScore is a recommended measure to assess the *informativeness* of generated summaries, while ROUGE-1 can effectively capture the *coherence* dimension. However, automatic and human scores for *conciseness* were negatively correlated, suggesting a potential direction for exploring this relationship in future work. Additionally, none of the metrics showed a strong correlation with human judgments of *fluency*, implying an open challenge.

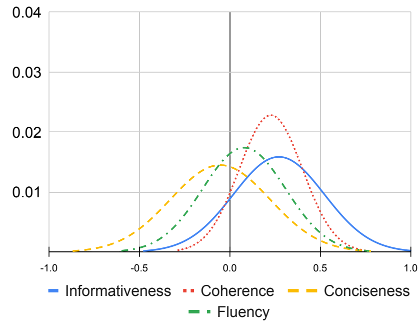
5.4. Supplemental Material Statement.

Our implementation is open source and can be accessed on the GitHub project.[8] We used the `transformer` library v4.25.1 from the `Huggingface hub` to implement our approach. We recommend following the official guideline[9] for setting up and loading the pre-trained language models (BART, $\text{BART}_{\text{large-CNN}}$ and T5).

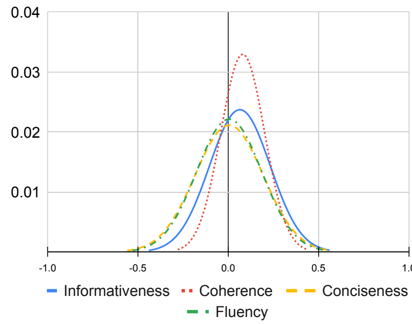
6. Conclusion

In this study, we explored using different language models for generating abstractive summaries of DBPEDIA abstracts. We observed that the existing shortened abstracts of DBPEDIA, which are obtained by truncating the full abstracts (i.e., selecting the first two-three sentences), may not cover all the relevant information. To overcome this limitation, we propose an abstractive summarization approach based on pre-trained language models such as BART and T5. We conducted various experiments on a multilingual dataset of DBPEDIA abstracts in six languages (English, Spanish, German, French, Italian, and Dutch). We employed automated metrics (Self-ROUGE, BERTScore) and human evaluation to investigate the best model for each language. Our results demonstrate that pre-trained language models can generate informative and concise summaries of DBPEDIA abstracts. However, selecting the most suitable model for each language is

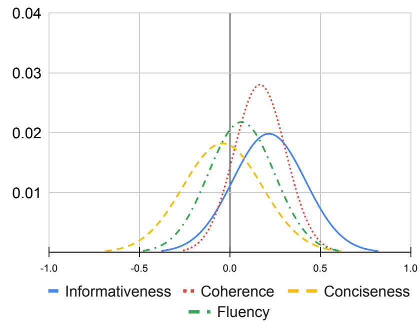
[8] https://github.com/dice-group/DBpedia-Summarizer
[9] https://huggingface.co/docs/transformers/index



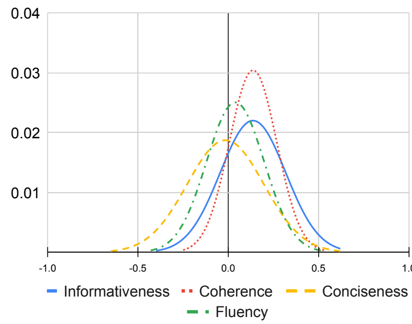
(a) Spearman's rank correlation coefficient for BERTScore



(b) Spearman's rank correlation coefficient for ROUGE-1



(c) Kendall's rank correlation coefficient for BERTScore



(d) Kendall's rank correlation coefficient for ROUGE-1

Figure 7. Correlation comparison between human and automated evaluations.

crucial. Furthermore, we found a correlation between automated and human evaluation for assessing *informativeness* with BERTScore and coherence with ROUGE-1. There is also a negative correlation for *conciseness* with human ratings. The evaluation of *fluency* is challenging without human involvement. We plan to investigate larger pre-trained language models in our future studies and fine-tune them on abstractive summarization datasets such as XL-Sum and Wikisum.

Acknowledgement

This work has been supported by the German Federal Ministry of Education and Research (BMBF) through the EuroStars project $E!114154$ PORQUE (grant no $01QE2056C$) and the KIAM project (grant no $02L19C115$). Additionally, this work has been partially supported by: the Department of Research and Universities of the Government of Catalonia ($SGR00930$), the Ministry of Science and Innovation of Spain with the project COMCRISIS (reference code $PID2019-109064GB-I00$), the EU-funded SoBigData++ project under Grant Agreement 871042 and MCIN/AEI /10.13039/501100011033 under the Maria de Maeztu Units of Excellence Programme ($CEX2021-001195-M$).

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
2. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.
3. Marius Kavaliauskas and Remigijus Venskutonis. Knowledge graphs in search engines: A systematic literature review. *Informatics*, 7(4):72, 2020.
4. Kaili Sun, Xudong Luo, and Michael Y. Luo. A survey of pretrained language models. In *Knowledge Science, Engineering and Management*, 2022.
5. Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. Pretrained language model for text generation: A survey. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4492–4499. ijcai.org, 2021. . URL https://doi.org/10.24963/ijcai.2021/612.
6. Zaid Lawal and Wei Lu. Evaluating the effectiveness of modern pre-trained language models for text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6479–6490. Association for Computational Linguistics, July 2020. URL https://www.aclweb.org/anthology/2020.acl-main.720.
7. Yaser Keneshloo, Mahdi Namazifar, and Hamed Zamani. Fine-tuning pre-trained transformer models for abstractive text summarization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 2723–2732. Association for Computing Machinery, October 2020. . URL https://doi.org/10.1145/3340531.3411917.
8. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.
9. Yichong Xu, Ruochen Xu, Dan Iter, Yang Liu, Shuohang Wang, Chenguang Zhu, and Michael Zeng. Inheritsumm: A general, versatile and compact summarizer by distilling from gpt, 2023.
10. Ayesha Ayub Syed, Ford Lumban Gaol, Alfred Boediman, Tokuro Matsuo, and Widodo Budiharto. A survey of abstractive text summarization utilising pretrained language models. In Ngoc Thanh Nguyen, Tien Khoa Tran, Ualsher Tukayev, Tzung-Pei Hong, Bogdan Trawiński, and Edward Szczerbicki, editors, *Intelligent Information and Database Systems*, pages 532–544, Cham, 2022. Springer International Publishing.
11. Yuntian Zhang, Hanjun Dai, Yiming Li, Zihang Liu, Jianfeng Gao, Jaime Carbonell, Caiming Xiong, and Ying Liu. Bart: Pre-training sequence to sequence models for language generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

12. Yuting Zhang, Ting Liu, Min Zhang, Yantao Chen, and Guilin Gao. Summarizing real-world entities in knowledge graphs: A survey. *Artificial Intelligence Review*, 52 (4):493–535, 2019.
13. Mohamed Elgharib, Jing He, Kukka Tero, Diana Inkpen, and Jimmy Chen. Automatic summarization of knowledge graph entities. In *Proceedings of the International Conference on Web Intelligence*, pages 600–607. ACM, 2017.
14. Avaneesh Kumar Yadav, Ranvijay, Rama Shankar Yadav, and Ashish Kumar Maurya. State-of-the-art approach to extractive text summarization: a comprehensive review. *Multimedia Tools and Applications*, pages 1–63, 2023.
15. Som Gupta and Sanjai Kumar Gupta. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65, 2019.
16. Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.
17. Shashi Narayan, Shay B Cohen, and Mirella Lapata. Fine-tuning pre-trained transformer models for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
18. Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. News article summarization with pretrained transformer. In *International Advanced Computing Conference*, pages 203–211. Springer, 2020.
19. Elena Lloret, Laura Plaza, and Ahmet Aker. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52:101–148, 2018.
20. Elena Lloret, Laura Plaza, and Ahmet Aker. Analyzing the capabilities of crowdsourcing services for text summarization. *Language resources and evaluation*, 47: 337–369, 2013.
21. Neslihan Iskender, Tim Polzehl, and Sebastian Möller. Best practices for crowd-based evaluation of German summarization: Comparing crowd, expert and automatic evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.eval4nlp-1.16.
22. Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.
23. Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online, June 2021. Association for Computational Linguistics. . URL https://aclanthology.org/2021.naacl-main.57.
24. Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018.

25. Mikhail Koroteev. On the usage of semantic text-similarity metrics for natural language processing in russian. In *2020 13th International Conference "Management of large-scale system development" (MLSD)*, pages 1–4, 2020. .
26. Yizhu Liu, Qi Jia, and Kenny Zhu. Length control in abstractive summarization by pretraining information selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6885–6895, 2022.
27. Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *CoRR*, abs/2111.01243, 2021. URL https://arxiv.org/abs/2111.01243.
28. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.
29. Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. Pre-trained language models and their applications. *Engineering*, 2022. ISSN 2095-8099. . URL https://www.sciencedirect.com/science/article/pii/S2095809922006324.
30. Ayesha Ayub Syed, Ford Lumban Gaol, Alfred Boediman, Tokuro Matsuo, and Widodo Budiharto. A survey of abstractive text summarization utilising pretrained language models. In *Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28–30, 2022, Proceedings, Part I*, pages 532–544. Springer, 2022.
31. GS Ramesh, Vamsi Manyam, Vijoosh Mandula, Pavan Myana, Sathvika Macha, and Suprith Reddy. Abstractive text summarization using t5 architecture. In *Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2021*, pages 535–543. Springer, 2022.
32. Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*, 2020.
33. Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.
34. Ankit Sahu and Sriram G Sanjeevi. Better fine-tuning with extracted important sentences for abstractive summarization. In *Proceedings of the International Conference on Communication, Control and Information Sciences (ICCISc)*, volume 1, pages 1–6. IEEE, 2021.
35. Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
36. Chin-Yew Lin and FJ Och. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir workshop*, 2004.
37. Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.
38. Jerome L Myers, Arnold D Well, and RF Lorch Jr. Research design and statistical analysis routledge. *New York.[Google Scholar]*, 2010.

39. Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
40. Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 workshop on automatic summarization*, pages 45–51, 2002.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230004

QALD-9-ES: A Spanish Dataset for Question Answering Systems

Javier SORUCO [a,1], Diego COLLARANA [a,b,2], Andreas BOTH [c,d,3] and Ricardo USBECK [e,4]

[a] *Universidad Privada Boliviana, Cochabamba, Bolivia*
[b] *Fraunhofer FIT, Sankt Agustin, Germany*
[c] *Leipzig University of Applied Sciences, Leipzig, Germany*
[d] *DATEV eG, Nuremberg, Germany*
[e] *Universität Hamburg, Hamburg, Germany*

Abstract. Knowledge Graph Question Answering (KGQA) systems enable access to semantic information for any user who can compose a question in natural language. KGQA systems are now a core component of many industrial applications, including chatbots and conversational search applications. Although distinct worldwide cultures speak different languages, the number of languages covered by KGQA systems and its resources is mainly limited to English. To implement KGQA systems worldwide, we need to expand the current KGQA resources to languages other than English. Taking into account the recent popularity that Large-Scale Language Models are receiving, we believe that providing quality resources is key to the development of future pipelines. One of these resources is the datasets used to train and test KGQA systems. Among the few multilingual KGQA datasets available, only one covers Spanish, i.e., QALD-9. We reviewed the Spanish translations in the QALD-9 dataset and confirmed several issues that may affect the KGQA system's quality. Taking this into account, we created new Spanish translations for this dataset and reviewed them manually with the help of native speakers. This dataset provides newly created, high-quality translations for QALD-9; we call this extension QALD-9-ES. We merged these translations into the QALD-9-plus dataset, which provides trustworthy native translations for QALD-9 in nine languages, intending to create one complete source of high-quality translations. We compared the new translations with the QALD-9 original ones using language-agnostic quantitative text analysis measures and found improvements in the results of the new translations. Finally, we compared both translations using the GERBIL QA benchmark framework using a KGQA system that supports Spanish. Although the question-answering scores only improved slightly, we believe that improving the quality of the existing translations will result in better KGQA systems and therefore increase the applicability of KGQA w.r.t. the Spanish language domain.

Keywords. Knowledge Graphs, Question Answering, Dataset

[1] Javier Soruco mail: javiersorucol1@upb.edu
[2] Diego Collarana mail: diego.collarana.vargas@iais.fraunhofer.de
[3] Andreas Both mail: andreas.both@datev.de
[4] Ricardo Usbeck mail: ricardo.usbeck@uni-hamburg.de

1. Introduction

The main goal of question-answering systems (QA systems) is to provide access to knowledge graphs via natural language, saving users from learning a specific graph query language to retrieve information from KGs. To achieve this goal, researchers have created different components and tools to mature the KGQA systems. These tools include benchmarking datasets to measure the quality of KGQA systems and datasets such as LC-QuAD [1] or QALD-9 [2] to train different KGQA components. Although natural language is the perfect medium for a comfortable experience for the end user, it also restricts who can take advantage of these systems. Recent developments in KGQA systems have heightened the need for multilingual tools and components. Ideally, KGQA systems should be available in various languages, making them accessible to diverse cultures. However, most KGQA research has focused mainly on English, leaving aside a significant number of languages, some of which are spoken by millions of people, e.g., Spanish, which is spoken by approximately 427 million people and is the world's second-most spoken native language[5].

QALD-9 is one of the few multilingual datasets that facilitate the development of KGQA systems in 11 languages. At the moment of writing this paper, QALD-9 is the only multilingual dataset available that provides Spanish translations. Unfortunately, most of the translations in QALD-9 are grammatically incorrect and unnatural[6]. Spanish is not the exception; after our analysis, we have found that the quality of Spanish translations of QALD-9 that have existed so far is relatively low. These issues go from poorly written translations to cases where the meaning of the original question is lost.

QALD-9-plus [3] has addressed this problem by improving the quality of these translations with the help of native speakers. QALD-9-plus adds translations in languages that were not included in the original benchmark, creating a dataset with high-quality translations available for nine different languages (en, de, fr, ru, uk, lt, be, ba, hy) and two knowledge graphs: Wikidata[7] and DBpedia[8].

In order to develop reliable KGQA systems for Spanish, the availability of high-quality resources that allow for training and testing of the systems becomes essential. We hope that improving the quality of a Spanish dataset will result in improved KGQA system performance for the given language. In this work, we aim to extend QALD-9-plus to include one additional language – Spanish. To achieve this goal, we manually created new translations with the help of native Spanish speakers. We also evaluated the results using language-agnostic quantitative text analysis measures and the tool GERBIL QA [4] to compare the results of the original translations and the new translations; we named the new translations "QALD-9-ES".

We address the problem of providing KGQA tools in multiple languages and propose QALD-9-ES, a KGQA dataset based on QALD-9 that contains accurate Spanish translations. We integrate QALD-9-ES with QALD-9-plus, complementing this multilingual dataset containing accurate natural translations with Spanish. Extending the scope of trustful translations for a dataset is essential for creating multilingual systems that serve a diverse population. In summary, the contributions of this work are as follows:

---

[5] cf. https://www.ethnologue.com/statistics/summary-language-size-19

[6] https://github.com/ag-sc/QALD/issues/22

[7] https://www.wikidata.org/

[8] https://www.dbpedia.org/

- A three-fold process to analyze KGQA datasets, i.e., Qualitative Analysis, Translations Review, and Quantitative Analysis. We apply this process to analyze the quality of the Spanish language in the QALD-9 dataset.
- A publicly available KGQA dataset with accurate Spanish translations. We integrate our work with QALD-9-plus to increase its adoption.
- An evaluation of our QALD-9-ES dataset against KGQA systems using the GERBIL QA framework and showing improvements in most of the metrics compared to its predecessor.

This article is organized into the following sections: (2) previous work, (3) dataset development and description, (4) baseline evaluation, and (5) conclusions.

2. Previous Work

For the elaboration of this work, we reviewed KGQA datasets and related tools to develop and compare KGQA datasets. Table 1 summarizes each dataset's information showing the lack of accurate Spanish translations.

Table 1. For existing KGQA datasets, we show the number of unique questions and available languages. QALD-9 is the only dataset available for the Spanish language, but it suffers from quality issues.

Dataset	Available for	No. of questions	Available languages
QALD-9	DBpedia	558	en, it, de, ru, fr, pt, hi_IN,fa, ro, es, nl
QALD-9-plus[9]	DBpedia Wikidata	558	en, lt, de, ru, fr, uk, be, ba, hy
rewordQALD9	DBpedia	551	en, it
LC-QUAD	DBpedia	5000	en
LC-QUAD 2.0	DBpedia Wikidata	30000	en

2.1. KGQA Datasets

2.1.1. QALD-9, the 9th Challenge on Question Answering over Linked Data (QALD-9):

QALD is a challenge with eleven years of history with the objective of providing up-to-date benchmarks for assessing and comparing state-of-the-art KGQA systems[10]. QALD-9 [2] is the 9th edition of the QALD challenge. This dataset provides 408 training questions and 150 test questions for DBpedia, available in 11 different languages, making QALD-9 one of the few multilingual KGQA benchmarks available and the only one we are aware of that counts with Spanish translations. In the 9th version of QALD, the questions were compiled and curated from previous versions and are accompanied by manually specified SPARQL queries and answers. The community reported multiple issues with the translations in the QALD-9 dataset; they were reported to be incorrect and of poor quality for different languages [3]. After reviewing the Spanish translations, we

[9] The number of questions differs depending on the language

[10] https://www.nliwod.org/challenge

found that Spanish was not the exception. We detail the issues found with the Spanish translations in Section 3.1. The QALD-9 dataset works with the QALD-JSON data format, which allows multiple languages, and it is used as a communication format with systems like GERBIL QA [4].

2.1.2. *QALD-9-plus, a Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers:*

QALD-9-plus [3] is an initiative to fight the lack of multilingual KGQA benchmarks and the translation issues of QALD-9. This dataset provides an extended version of QALD-9 with 4,930 new question translations for different languages. The translations were done via crowdsourcing. Each crowd worker was assigned a subset of QALD-9 questions to translate into their mother tongue, resulting in at least two translations per question. Crowd workers were later given two translations and the original question, and they had to decide whether the first or second translation was correct or whether both or no translations were correct. The QALD-9-plus dataset also includes a version of the dataset for Wikidata that was generated manually by three experienced computer scientists with the help of semi-automatic scripts to speed up the process. The result is an extended version of the QALD-9 dataset available for nine languages (English included) with high-quality translations, adjusted to work with both DBpedia and Wikidata knowledge graphs.

2.1.3. *RewordQALD9, a Bilingual Benchmark with Alternative Rewordings of QALD Questions:*

rewordQALD9 [5] is an extended version of the QALD-9 dataset that brings forward high-quality Italian translations with multiple reformulations for the same question. Rewordings are available for both Italian and English; therefore, testing systems' robustness is available for both languages. The translations were manually curated by native speakers, including reformulations for both English and Italian. The resulting dataset consists of 551 questions in both English and Italian. In addition, multiple question reformulations are included, i.e., 1546 for English and 1707 for Italian.

2.1.4. *LC-QuAD, a Corpus for Complex Question Answering over Knowledge Graphs:*

LC-QuAD [1] is the solution to the necessity of large datasets composed of various question templates and their logical forms for QA systems. LC-QuAD is generated based on an entity seed list and a predicate whitelist to obtain subgraphs from DBpedia. Then the graphs are used to generate a SPARQL from a template, which generates a natural language question from Normalized Natural Question Templates. Finally, the questions are manually reviewed and corrected. LC-QuAD is composed of 5000 questions, and the SPARQL queries are required to answer the questions on DBpedia. It was one of the most extensive datasets available for KGQA at the time of release. This dataset is only available in English; therefore, it is not multilingual.

2.1.5. *LC-QuAD 2.0, a Large Dataset for Complex Question Answering over Wikidata and DBpedia:*

LC-QuAD 2.0 [6] is the second version of LC-QUAD, providing 30,000 questions with their paraphrases and corresponding SPARQL queries. LC-QuAD 2.0 is compatible with Wikidata and DBpedia 2018 knowledge graphs. To generate the dataset, the authors generated SPARQL queries based on templates, which were then transformed into template questions. By using crowdsourcing, the template questions were verbalized into natural-language questions. This dataset, like its predecessor, is only available in English.

2.2. Tools Related to KGQA Systems and Benchmarks

2.2.1. Benchmarking Question Answering Systems:

GERBIL QA [4] is an online benchmarking platform for question-answering systems (derived from the GERBIL tool for evaluating entity recognition approaches, cf. [7]). GERBIL QA follows the FAIR principles to provide a quality evaluation of QA systems. This platform allows users to benchmark their systems with relevant datasets such as QALD and LC-QuAD. GERBIL QA also allows its users to upload their datasets. The platform is connected to relevant KGQA solutions so that the users can compare their systems with the relevant systems available; these systems can work with any private dataset uploaded by the users only if this dataset uses the QALD-JSON format. GERBIL QA offers seven metrics for benchmarking QA systems and supports online and file-based systems.

2.2.2. Question Answering Benchmark Curators

QUANT [8] is a framework for creating or curating QA benchmarks, generating smart edit suggestions for questions-query pairs and their metadata, and providing predefined quality checks for queries. QUANT reduces the curation effort for QA benchmarks by up to 91%. QUANT is a suitable tool during the KGQA dataset development process, e.g., QALD-9 used QUANT.

3. QALD-9-ES Dataset Development and Description

3.1. Qualitative Analysis of QALD-9

The QALD-9 dataset comprises 558 questions, of which 408 correspond to the training dataset and 150 to the testing dataset. Each question contains a list of question objects for every available language. A question object is composed of the question string and the question keywords. The question string is the question expressed in a given language (e.g., Spanish), and the question's keywords are key elements of the question that the KGQA system can use as support to answer questions for the given language. These keywords are usually related (but not restricted) to proper entity names, verbs, nouns, and adjectives.

As we mentioned before, QALD-9 already has Spanish translations; after an explorative review, we concluded that the quality of the Spanish translations is doubtful. Thus, we decided to review all the original translations, classifying each translation into seven cases.

Cases 1 to 6 correspond to error cases, which can be split into cases with errors in the question string (cases 1, 2, 5, and 6), or cases with mistakes in the question keywords (cases 3, 4), while case 7 implies a correct translation.

Each question can be assigned to multiple cases, with the only exception being case 7. If a sentence follows into Case 7, it implies a correct translation; therefore, it cannot be assigned to questions with translation errors in the question string, but it can be assigned to questions with mistakes in the question keywords.

1. The first case involves what we call "minor translation mistakes," which we define as translation errors that do not alter the question's original meaning. Some common errors, in this case, are the use of the incorrect genders (for example, using the word "hermoso" for a female noun or subject), missing words that do not contribute to the meaning of the question, the absence of opening and closing question marks ('¿?'), missing letters, the inappropriate use of plural forms, the lack of capital letters in proper names, or using the wrong tense. An example of these issues can be found within the question "Who developed Skype?", which got translated to "Quien desarrollado Skype?"; The verb "desarrollado" is in its past participle form. To be a correct translation, the auxiliary verb "ha" should accompany the main verb resulting in the translation "¿Quién ha desarrollado Skype?". Another correct alternative is to modify the main verb by changing it to the simple past tense, resulting in "¿Quién desarrolló Skype?".
2. The second case is about what we call "major translation errors", the main characteristic of these errors is that they result in the loss of meaningful elements of the question, such as verbs, proper names, and other meaningful words. One example of this case is found in the question "In which U.S. state is Area 51 located?" which got translated to "En cual Nosotros estado es Zona 51 ¿situado?". If we reverse this translation, we would get "In which us state is Zone 51 located?" As you can see, the entity "U.S." was mistranslated to the word "us" and the entity "Area 51" was transformed to "Zone 51". In both cases, the entities in question were lost, meaning that a KGQA system would not be able to work properly with this translation. Another example of this issue can be found in the question "Where did Abraham Lincoln die?" that gets translated to "Dónde hizo Abrahán Lincoln el?". In this translation, the verb "die" was completely lost, resulting in a question that just does not make sense. In the original QALD-9 translations, we found some questions that lacked their corresponding Spanish translation. Those questions were classified into this case. This case can be triggered by mistakes both in the question and in the question's keywords.
3. The third case relates to questions where the question's keywords require modifications due to some mistakes in the question's original translation that propagate into the question's keywords. We can find this case in the question "What is the last work of Dan Brown?", which got translated to "Qué es el último trabajo de Y ¿Marrón?", In this case, the entity Dan Brown was mistranslated to "Y ¿Marrón?", resulting in error propagation into the question keywords "último trabajo, Y marrón". The correct keywords for this question are "último trabajo, Dan Brown".
4. The fourth case is similar to the third case; the difference is that the errors found in the question's keywords are not related to errors in the question's translation. For example, words are correctly written in the question's translation but wrongly written in the question's keywords. In the question "Which monarchs were married to a German?", we find that the Spanish keywords for the question are "monarcha, casado, alemán". The word "monarcha" is wrongly translated; the correct translation "monarca". This mistranslation was only found in the question's keywords; the translation lacks this mistake.

5. The fifth case is about Spanish accentuation. In Spanish, the character "´ " is known as "orthographic accent". This accent is used on some words' vowels and can modify the word's meaning. One example is the words "mas" and "más". "mas" is an equivalent of the word "but", while "más" is a quantity adverb. Some QALD-9 translations lack orthographic accents, so the real meaning of some words is lost. This case can be triggered by mistakes both in the question and in the question keywords.
6. The sixth case comprises questions with correct translations that were reformulated to be more natural. Modifications in the question's keywords do not trigger this case. The question "How deep is Lake Placid?" was translated to "Cómo de hondo es el Lago Placid?", which is technically a correct translation, but we consider that the literal translation "Cómo de hondo" is not the best way to express "How deep"; therefore, we modified the translation to "¿Qué tan profundo es el Lago Placid?".
7. Finally, the seventh case relates to questions that do not require modification. Modifications to the question's keywords do not affect this case. The question "Which presidents were born in 1945?" was translated to "¿Qué presidentes nacieron en 1945?". We consider this translation correct and natural; therefore, we classified it under Case 7.

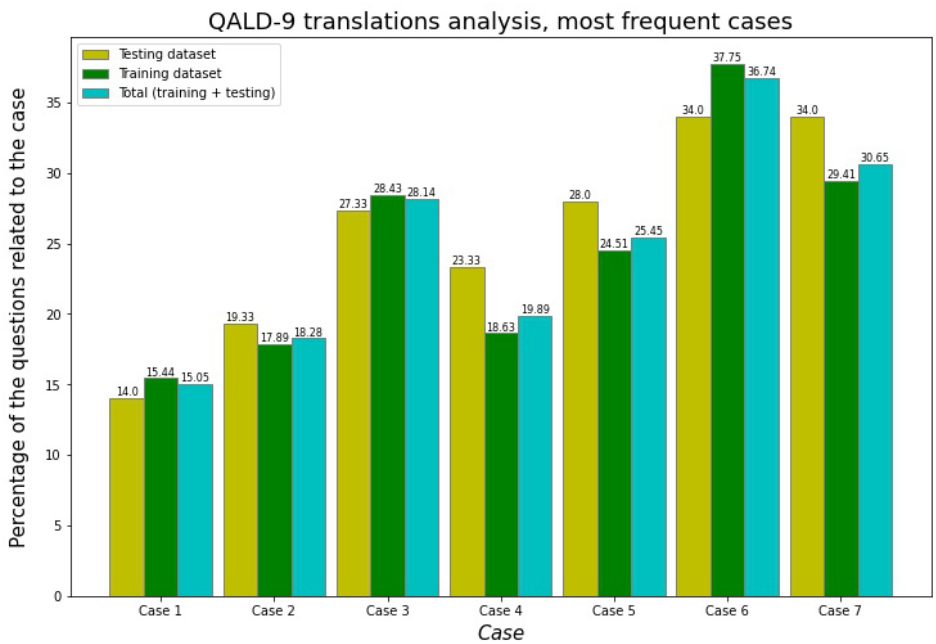


Figure 1. Plot showing the percentage of occurrence of each case in the QALD-9 dataset. Case 1 is related to questions with minor translation issues, case 2 to major translation issues resulting in the loss of the original meaning of the question, case 3 relates to errors on the question keywords propagated from the question translation, case 4 to mistakes in the question keywords that are not related to errors in the question translation, case 5 is about questions with accentuation errors, case 6 contains questions that were modified to be more natural, and finally, case 7 are questions with correct translations.

Figure 1 shows the percentage of occurrence of each case, 15.05% of the question presented minor translation mistakes (Case 1), 18.28% of the questions had major translation issues (Case 2), 28.14% of the question's keywords had mistakes that resulted from errors in the question's translation (Case 3). 19.89% of the question keywords had errors that were not related to errors in the question's translation (Case 4), 25.45% of the question presented accentuation errors in the question string or the question keywords (Case 5), 36.74% of the questions were modified due to their lack of naturalness (Case 6), and 30.65% of the questions were considered correct translations. We also found 23 questions lacking Spanish translation; these were classified as major translation issues.

3.2. Translation of Questions

The translation process consisted of two agents, "the translator" and "the reviewer", interacting within two stages: the translation stage and the review stage.

The translator is a native Spanish speaker with the main objective of generating new translations while matching the old translations to one of the seven cases described before. This agent is also required to have a high level of understanding of SPARQL and the QALD-JSON format in order to be able to check the question information if required.

The reviewer is a native Spanish speaker with the main objective of reviewing the new translation and providing feedback to the translator to improve the quality of the new translations.

In the translation stage, the translator reviews each question by checking the original English question, its original QALD-9 translation, the English question keywords, and the QALD-9 translation keywords. The translator annotates the cases that describe the question translation (see above) and generates a translation for the English question. The translation is kept if the question's translation is correct (case 7). The translator will always look for the Spanish version of proper names in order to keep the translation as natural as possible (cases like "Iraq", which is spelled "Irak" in Spanish); this also implies not translating proper English names if the Spanish-speaking community knows the entity by the English name (like the TV show "Friends"). The translator's goal is always to generate correct translations (avoid grammatical and accentuation mistakes) that fit in the natural Spanish dialect.

In the review stage, the reviewer checks the original English question, the translation generated by the translator, and the keywords from the new translation, looking for possible mistakes. The mistakes are annotated and sent to the translator. The translator reviews the observations, and if the translator agrees with the correction, the question is modified; corrections that raise additional concerns about the translation are annotated and discussed with the reviewer. The reviewer explains why the corrections are necessary so the translator can make a choice; in some cases, the translator reviews the question's SPARQL query and answers to make an appropriate choice. This process was executed over the original QALD-9 dataset. Once the native translations were generated, they were merged into QALD_9_plus using the question's id, generating what we call QALD-9-ES[11].

[11] https://github.com/KGQA/QALD_9_plus

Table 2. Number of unique questions available for Wikidata and DBpedia in the QALD-9-plus extension QALD-9-ES.

	en	de	ru	uk	lt	be	ba	es	hy	fr
DBpedia train	408	543	1203	447	468	441	284	408	80	260
DBpedia test	150	176	348	176	186	155	117	150	20	26
Wikidata train	371	497	1095	407	426	403	260	371	71	251
Wikidata test	136	159	318	160	166	141	107	136	19	25

Table 3. Results of the linguistic evaluation of QALD-9 and QALD-9-ES Spanish translations performed by using LinguaF. The linguistic indicators used to compare the translations are average words per sentence, average word length, average syllable per word, lexical density, and type-token ratio.

	QALD-9 Train	QALD-9-ES Train	QALD-9 Test	QALD-9-ES Test
Average words per sentence	7.044706	7.504808	6.767442	7.434211
Average word length	5.034402	5.097053	5.058419	5.114159
Average Syllable per word	1.418838	1.426650	1.416667	1.417699
Lexical density	78.423514	79.724536	76.804124	79.292035
Type Token Ratio	0.376420	0.356502	0.472509	0.465487

3.3. Dataset Statistics

The resulting dataset contains questions for DBpedia and Wikidata Knowledge Graphs. In Table 2, we show that the dataset contains 408 train questions for DBpedia, 371 train questions for Wikidata, 150 test questions for DBpedia, and 136 test questions for Wikidata in Spanish. The rest of the languages are preserved as in QALD-9-plus [3].

3.4. Quantitative Analysis between QALD-9 vs QALD-9-ES

Inspired by the work of the QALD-9-plus team, we have used language-agnostic quantitative text analysis measures to observe the differences between the Spanish translations of QALD-9 and QALD-9-ES. To achieve this, we used the library “LinguaF”[12].

The results can be observed in Table 3. QALD-9-ES' translations have more words; each word is longer and has more syllables. QALD-9-ES also improves the dataset's lexical density, meaning that the dataset has more meaningful words (e.g., nouns, verbs, adjectives, and some adverbs). On the other hand, we have found that the new translations present a lower Type Token Ratio (TTR). That means there are fewer unique words in the new translations than in the old ones. After analyzing and comparing both translations, we hypothesized that this was because most questions that presented translation errors (like Case 1 and 3) tended to add incorrect and unrelated words. Questions related to Case 5 affect this measure too, as words that have errors related to accentuation in some questions are considered unique words by LinguaF.

[12] https://github.com/WSE-research/LinguaF

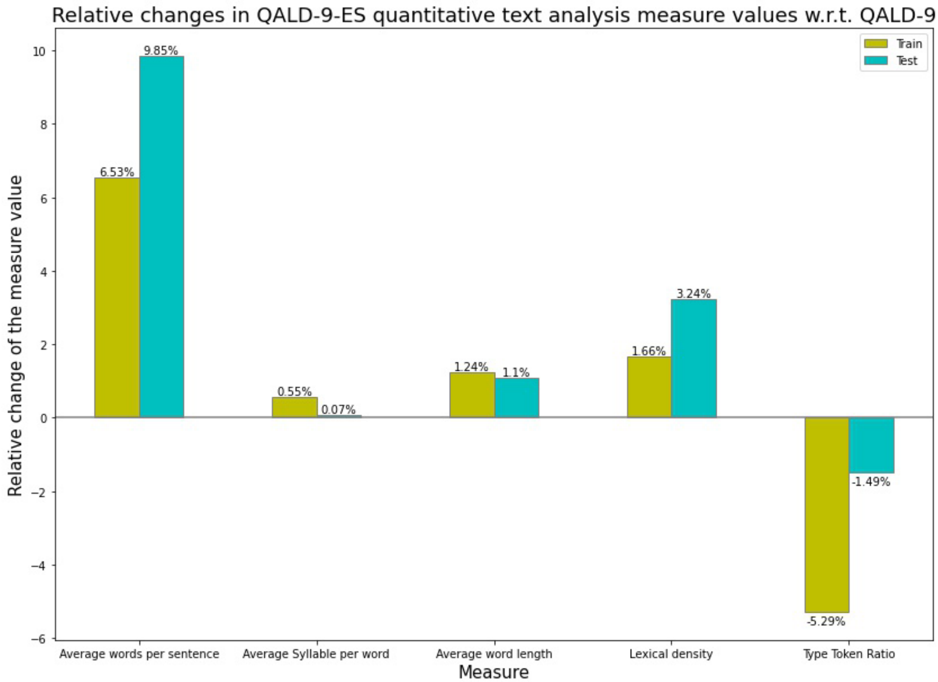


Figure 2. We use LinguaF to perform the linguistic evaluation, showing relative improvements in the Spanish translations comparing QALD-9 vs. QALD-9-ES. The linguistic indicators used are average words per sentence, average word length, average syllable per word, lexical density, and type-token ratio.

In Figure 2, we can see that the measures with the most improvement in QALD-9-ES are average words per sentence and lexical density, followed by average word length and average syllable per word, making TTR the only measure with a decrement in comparison to QALD-9.

4. Baseline Evaluation

With the premise that better translations result in better QA systems, we used the GERBIL QA system to evaluate the QALD-9-ES dataset by comparing it to QALD-9. At the time of writing this paper, QAnswer [9] is the only working annotator available on GERBIL QA that supports Spanish QA. QAnswer is available in two versions, one that works with DBpedia KG and the other with Wikidata KG. While working with DBpedia, both QALD-9-ES and QALD-9 resulted in system errors; therefore, we only compared the datasets using QAnswer over Wikidata. QALD-9 does not include a version for Wikidata; hence, we created a version by replacing the QALD-9-plus translations with the original QALD-9 translations in the Wikidata set.

When running the experiments on GERBIL QA, we received not only the QA results but the results of three sub-experiments that can measure the quality of a QA system; Resource to Knowledge Base (C2KB), Properties to Knowledge Base (P2KB) and Relation to Knowledge Base (RE2KB). The results for QALD-9-ES are shown in Table 5, and for QALD-9 in Table 4. For each sub-experiment, we present the F1 score metric.

Table 4. Results of the GERBIL QA evaluation performed on QALD-9 Spanish, using the QAnswer annotator over Wikidata. GERBIL performs four sub-experiments: Question Answering (QA), identification of relevant resources (C2KB), identification of relevant properties (P2KB), and matching of expected triples (RE2KB).

Dataset	Sub-experiment	Micro F1	Macro F1
QALD-9 test	QA	0.1077	0.1522
	C2KB	0.3798	0.3408
	P2KB	0.4183	0.4069
	RE2KB	0.1141	0.1691
QALD-9 train	QA	0.1588	0.2486
	C2KB	0.3682	0.3538
	P2KB	0.4082	0.4106
	RE2KB	0.1703	0.2276

Table 5. Results of the GERBIL QA evaluation performed on QALD-9-ES using the QAnswer annotator over Wikidata. GERBIL performed four sub-experiments: Question Answering (QA), identification of relevant resources (C2KB), identification of relevant properties (P2KB), and matching of expected triples (RE2KB). We see improvements in most of the scores when using QALD-9-ES.

Dataset	Sub-experiment	Micro F1	Macro F1
QALD-9-ES test	QA	0.1142	0.1680
	C2KB	0.3893	0.3593
	P2KB	0.4093	0.3951
	RE2KB	0.1236	0.1863
QALD-9-ES train	QA	0.1775	0.2639
	C2KB	0.3623	0.3471
	P2KB	0.3804	0.3887
	RE2KB	0.1874	0.2419

The F1 score is the harmonic mean between the system's precision and recall presented in Eq. (1). This metric is applied over one class; in this case, we have several classes; therefore, the micro and macro average aggregation methods are applied. Macro F1 is the unweighted mean of the F1 scores obtained per class presented in Eq. (2). The Micro F1 method calculates the F1 score using the normal F1 equation but using the total number of True Positive (TP), False Positive (FP), and False Negative (FN) values instead of the values of a single class and it is presented in Eq. (3). The main difference between the micro and macro metrics is that Micro F1 gives equal importance to each observation; consequently, some classes will significantly impact the results for imbalanced datasets. On the other hand, Macro F1 gives equal importance to the class F1 score, allowing it to return objective results even on imbalanced datasets.

$$F1 = \frac{TP}{TP + \frac{1}{2} * (FP + FN)} \tag{1}$$

$$Macro\,F1 = \frac{\sum F1\ scores}{Number\ of\ classes} \tag{2}$$

$$Micro\,F1 = \frac{\sum TP}{\sum TP + \frac{1}{2} * (\sum FP + \sum FN)} \tag{3}$$

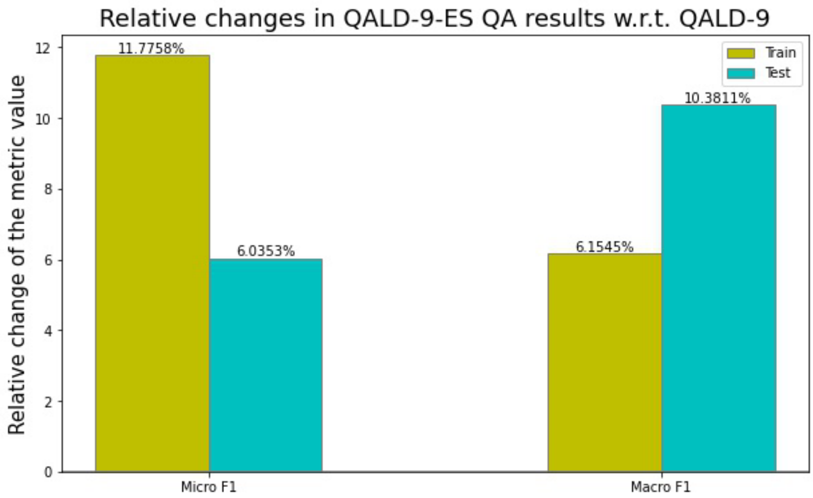


Figure 3. Relative changes in the question answering (QA) results of QAnswer over Wikidata using the QALD-9-ES Spanish translations compared to the same annotator using the QALD-9 Spanish translations.

The results of the QA experiment show an increment in both Micro and Macro F1 (Figure 3) when using QALD-9-ES. The Micro F1 measure value increased more in the training set than in the testing set, and Macro F1 shows a more significant improvement in the testing set than in the training set.

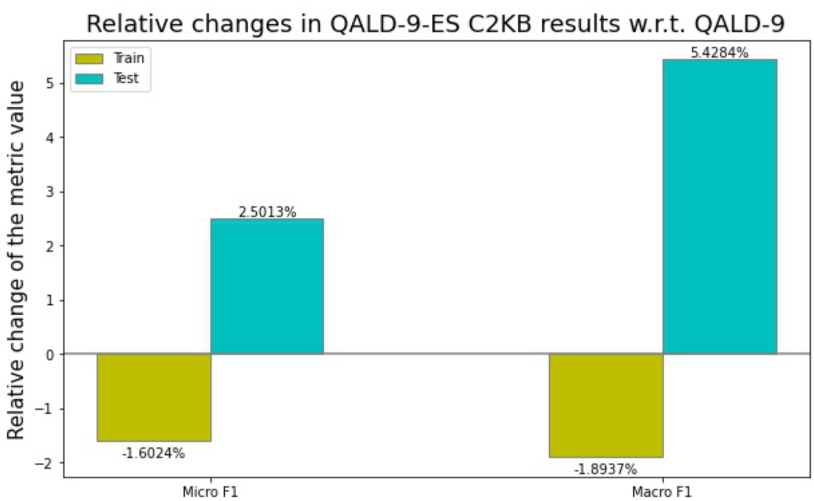


Figure 4. Relative changes in the GERBIL sub-experiment C2KB results of QAnswer over Wikidata using the QALD-9-ES Spanish translations compared to the same annotator using the QALD-9 Spanish translations.

The C2KB sub-experiment qualifies the capability of the system to identify all the relevant resources for the given question. Figure 4 shows a relative increment in Micro and Macro F1 scores for the testing set, but the scores decrement for the training set.

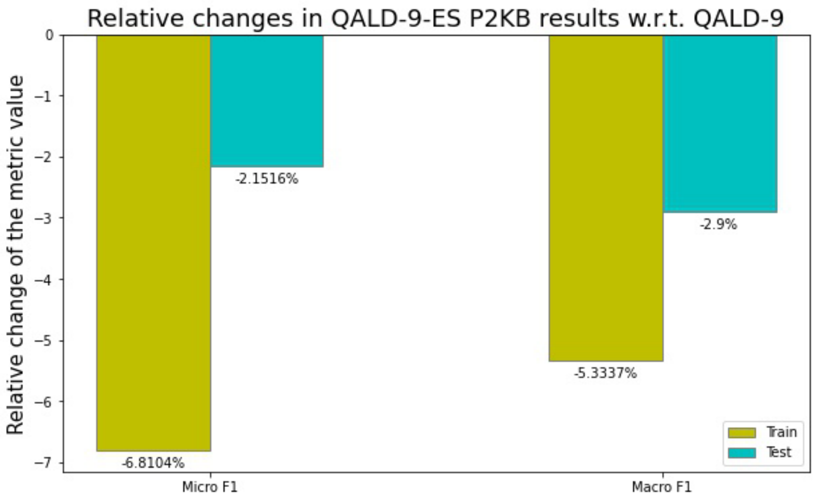


Figure 5. Relative changes in the GERBIL sub-experiment P2KB results of QAnswer over Wikidata using the QALD-9-ES Spanish translations compared to the same annotator using the QALD-9 Spanish translations.

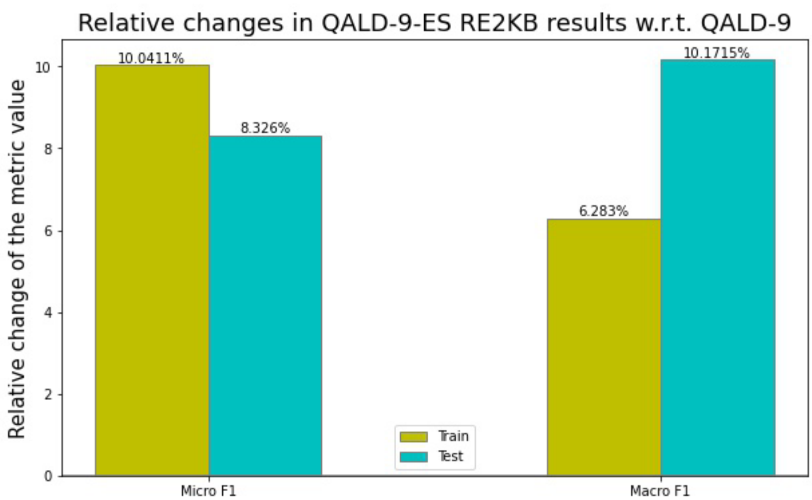


Figure 6. Relative changes in the GERBIL sub-experiment RE2KB results of QAnswer over Wikidata using the QALD-9-ES Spanish translations compared to the same annotator using the QALD-9 Spanish translations.

The P2KB sub-experiment qualifies the system's capability to identify all the relevant properties for the given question. In this case, we see a decrement for Micro and Macro F1 for both the testing and training datasets (Figure 5). Finally, the RE2KB sub-experiment compares the expected triples in the question's expected SPARQL against the triples in the SPARQL returned by the QA system. QALD-9-ES shows an improvement in this sub-experiment compared to QALD-9 in both the testing and training sets for Micro and Macro F1 (cf. Figure 6).

5. Conclusions

KGQA systems provide access to knowledge graph information through natural language. However, the number of unique natural languages is not comparable to the number of languages covered by existing KGQA systems. This paper addresses the problem of providing multilingual tools to develop KGQA systems to increase the number of languages these systems cover. We focus on Spanish, a language spoken by more than 450 million people worldwide.

Following a three-fold approach, we performed a qualitative analysis of the Spanish translations presented in QALD-9 and found that only 30.65% of the Spanish translations in QALD-9 properly represent the English questions. Then, manually generate new translations for the questions that presented translation issues and review them manually with the help of native speakers. We integrate QALD-9-ES with QALD-9-plus, a QALD-9-based dataset made exclusively with native translations, so there is a complete source of high-quality translations for QALD-9 that can be used for the development of new datasets and KGQA systems.

We compared the QALD-9-ES Spanish translations with the original translation included in QALD-9 using language-agnostic quantitative text analysis measures to confirm that the new translations use more words, each word is longer, and there are more meaningful words in each translation. The only downside is that there are fewer unique words. After some review, we hypothesize that this result is because the original QALD-9 translations often included unrelated words in the translations and several accentuation mistakes that are taken as unique words.

Using the GERBIL QA framework, we also evaluate both datasets, i.e., QALD-9-ES vs QALD-9. We use the QAnswer annotator for Wikidata (the only annotator working in Spanish KGQA at the time of evaluation). The experiment results show minor improvements in the QA results. The system also showed slightly better results in identifying relevant resources (C2KB) for the testing set and slightly worse results for the training set. We also found that the annotator has lower performance in identifying the relevant properties for a given question (P2KB) with the new translations, but the annotator has better results in matching the expected triples for each question (RE2KB). These mixed results show the impact of the quality of the dataset on the KGQA system and components.

Finally, we demonstrated how the QALD-9-ES dataset is useful for the development of Spanish KGQA pipelines. We expect that this new dataset will especially benefit KGQA systems that use Large-Scale Language Models in their pipeline. The resulting dataset of this work was merged into QALD-9-plus, a fully native translated dataset, this resource can also be used to compare native translations against translations generated using Machine Translation.

Acknowledgement: This work is supported by the Research Partnership Grant RPG2106 funded by the Swiss Leading House for Latin America, and by grants for the DFG project NFDI4DataScience project (DFG project no. 460234259) and by the Federal Ministry for Economics and Climate Action in the project CoyPu (project number 01MK21007G).

References

[1] Trivedi P, Maheshwari G, Dubey M, Lehmann J. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In: The Semantic Web - ISWC 2017 - 16th International Semantic Web

Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II. vol. 10588 of Lecture Notes in Computer Science. Springer; 2017. p. 210-8.

[2] Ngomo N. 9th challenge on question answering over linked data (QALD-9). vol. 7; 2018. p. 58-64.

[3] Perevalov A, Diefenbach D, Usbeck R, Both A. QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers. In: 2022 IEEE 16th International Conference on Semantic Computing (ICSC); 2022. p. 229-34.

[4] Usbeck R, Röder M, Hoffmann M, Conrads F, Huthmann J, Ngomo AN, et al. Benchmarking question answering systems. Semantic Web. 2019;10(2):293-304.

[5] Sanguinetti M, Atzori M, Puddu N. rewordQALD9: A Bilingual Benchmark with Alternative Rewordings of QALD Questions. In: Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems co-located with 18th International Conference on Semantic Systems (SEMANTiCS 2022), Vienna, Austria, September 13th to 15th, 2022. vol. 3235 of CEUR Workshop Proceedings. CEUR-WS.org; 2022. .

[6] Dubey M, Banerjee D, Abdelkawi A, Lehmann J. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In: The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II. vol. 11779 of Lecture Notes in Computer Science. Springer; 2019. p. 69-78.

[7] Usbeck R, Röder M, Ngonga Ngomo AC, Baron C, Both A, Brümmer M, et al. GERBIL: General Entity Annotator Benchmarking Framework. In: Proceedings of the 24th International Conference on World Wide Web. WWW '15. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee; 2015. p. 11331143. Available from: https://doi.org/10.1145/2736277.2741626.

[8] Gusmita RH, Jalota R, Vollmers D, Reineke J, Ngomo AN, Usbeck R. QUANT - Question Answering Benchmark Curator. In: Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9-12, 2019, Proceedings. vol. 11702 of Lecture Notes in Computer Science. Springer; 2019. p. 343-58.

[9] Diefenbach D, Giménez-García J, Both A, Singh K, Maret P. QAnswer KG: Designing a Portable Question Answering System over RDF Data. In: Harth A, Kirrane S, Ngonga Ngomo AC, Paulheim H, Rula A, Gentile AL, et al., editors. The Semantic Web. Cham: Springer International Publishing; 2020. p. 429-45.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230005

Towards a Versatile Terminology Service for Empowering FAIR Research Data: Enabling Ontology Discovery, Design, Curation, and Utilization Across Scientific Communities

Philip Strömert[a,1][0000-0002-1595-3213], Vatsal Limbachia[a][0000-0002-9535-6960], Pooya Oladazimi[a][0000-0002-5117-070X], Johannes Hunold[a][0000-0002-4378-6061], and Oliver Koepler[a,1][0000-0003-3385-4232]
[a]*TIB - Leibniz Information Centre for Science and Technology, Hannover, Germany*[04aj4c181]

Abstract. To fully harness the potential of data, the creation of machine-readable data and utilization of the FAIR Data Principles is vital for successful data-driven science. Ontologies serve as the foundation for generating semantically rich, FAIR data that machines can understand, enabling seamless data integration and exchange across scientific disciplines. In this paper, we introduce a versatile Terminology Service that supports various tasks, including discovery, provision, as well as ontology design and curation. This service offers unified access to a vast array of ontologies across scientific disciplines, encouraging their reuse, improvement, and maturation. We present a user-driven service development approach, along with a use case involving a collaborative ontology design process, engaging domain experts, knowledge workers, and ontology engineers. This collaboration incorporates the application and evaluation of the Terminology Service, as well as supplementary tools, workflows, and collaboration models. We demonstrate the feasibility, prerequisites, and ongoing challenges related to developing Terminology Services that address numerous aspects of ontology utilization for producing FAIR, machine-actionable data.

Keywords. FAIR Data, Ontology, Terminology Service, Research Data, Chemistry

1. Introduction

Resources like re3data [1] or FAIRsharing [2] provide comprehensive collections of data repositories, databases, data, and metadata standards and policies. Initiatives like the European Open Science Cloud (EOSC) [3], the German National Research Data Infrastructure (NFDI) [4], and the Physical Sciences Data Infrastructure (PSDI) in the United Kingdom [5] create and provide services and infrastructures to make research data publicly available, adapting the FAIR data principles [6]. The vision of Open

[1] Corresponding Authors: Philip Strömert, philip.stroemert@tib.eu; Oliver Koepler, oliver.koepler@tib.eu.

Science has gained momentum and is being put into practice by more and more scientific communities.

The availability of increasingly large amounts of data enables the exploration of new data-driven, interdisciplinary research questions. These approaches require integration and harmonization of machine-actionable data across disciplinary boundaries addressing Findability, Accessibility, Interoperability, and Reusability. The idea of having machine-actionable data is derived from the JDDCP [7] guideline that refers to making the data readable and modifiable by machines.

Machine-actionable, FAIR research data is achieved by annotation with rich metadata. Furthermore, these metadata themselves also need to be FAIR, meaning metadata used for data annotation needs to be understandable and actionable by both humans and machines. Generic metadata schema like DataCite [8] or Dublin Core [9] cannot express rich discipline-specific descriptions of data. This often results in limiting the annotation to basic metadata such as title, author, date, and format and providing the domain-specific metadata in the form of long free text that decreases data FAIRness due to the lack of semantics [10].

Here, terminologies play an important role in the creation of semantically rich, discipline-specific metadata. They further provide the basis for consensus definitions of entities, thereby ensuring conceptual alignment across domains, even when the nomenclature differs between domains. The use of metadata schemata implementing standardized terminologies promotes interoperability and data integration, as data described through common terminologies can be understood and used across different systems and disciplines.

Terminology services like the Ontology Lookup Service [11], Bioportal, or Linked Open Vocabularies provide access to either general or discipline-specific collections of ontologies, terminologies, or vocabularies. They offer features like browsing, searching, filtering, and downloading ontologies. Terminology services, therefore, play a crucial role in the identification of relevant ontologies in the process of data annotation and the creation of FAIR data. Their relevance is reflected in the integration into various data annotation or data management tools. The CEDAR workbench supports data annotation with domain-specific metadata schema [12]. For the creation of metadata templates, the workbench integrates ontology terms from the Bioportal terminology service [13] which the user can select from. The Dataverse project enables the integration of customized metadata schema which can be populated with terms of linked terminology services [14] [15]. The electronic lab notebook Chemotion utilizes terminologies to annotate data of experiments [16].

With the increasing use of terminologies for data annotation, errors and gaps in terminologies are inevitably uncovered not only by ontology engineers but also by domain experts annotating data. This opens up new application areas for terminology services. What could be more natural than using the services that are used for searching and analyzing ontologies for curation and development as well?

In this paper, we discuss collaborative ontology development workflows and the requirements for a versatile Terminology Service to support these workflows. What are the best approaches and best practices to collaboratively design and curate ontologies in a team of ontology engineers, knowledge workers, and domain experts? What are the requirements for a Terminology Service to support the design and curation workflows of stakeholders with varying levels of expertise? How can a Terminology Service improve ontology development concerning harmonizing and aligning the application and reuse of terminologies across knowledge domains?

In the following section, we provide an overview of related work concerning terminology services and ontology development tools. In the subsequent section, we briefly describe the challenges when using terms from terminologies for data annotation or when reusing terms from existing terminologies in new ones. Section 4 describes our approach to collaborative ontology development and lessons learned in the application of a terminology service for the derived workflows. In section 5 we describe our terminology service and the enhancements we have developed based on the requirements derived from experiences and observations during the ontology development process. Section 6 summarizes and discusses insights and evaluation of user-driven development processes to extend the terminology service as a tool for ontology development and curation.

2. Related Work

Two types of services and tools aim to facilitate ontology access and development: Terminology Services (TS) and ontology development tools. In this section, we present the most prominent open-source tools and services and also discuss deficiencies and gaps in existing solutions concerning envisioning a well-integrated TS for both ontology development and data annotation.

2.1 Terminology Services (TS)

A number of mature terminology services and repositories are publicly available. Terminology registries such as the Basic Register of Thesauri, Ontologies & Classifications (BARTOC) represent the simplest form of such a service [17]. Registries usually list comprehensive metadata about vocabularies, terminologies, and ontologies and link to their original source while Terminology repositories provide access to the terminology data itself, often in combination with access to this data via APIs [18]. Well-used TS frameworks like OntoPortal [19] and Ontology Lookup Service (OLS) [20] provide extended features like searching and browsing within the indexed terminologies, tree views of concepts and properties, and visualization. OntoPortal is an ontology-supported portal architecture developed by the OntoPortal Alliance that can be customized to provide discipline-specific terminology services. It has emerged from BioPortal, which provides access to a collection of biomedical ontologies and terminologies [13]. The Ontology Lookup Service, developed and hosted at the EMBL European Bioinformatics Institute (EBI), provides ontology search and visualization services as well as data access and search through an API. Furthermore, there is Skosmos, an open-source web-based browser and publishing tool specialized for Simple Knowledge Organization System (SKOS) vocabularies [21]. It provides a user interface for browsing and searching the data as well as Linked Data access with APIs that support term-based searches. DBpedia Archivo archives and ontologies on a web-scale to offer access to their different versions over time. It also provides quality metrics [22]. The Linked Open Vocabularies (LOV) portal provides a comprehensive overview of vocabularies in the realm of the Semantic Web and Linked Data, which can also be accessed via a SPARQL endpoint [23].

All terminology services provide rich functionalities, yet they typically encompass ontologies from specific domains or disciplines alone. While this approach proves feasible in many instances where users want to focus on their respective domains, it does pose challenges for those seeking a comprehensive overview of available ontologies across diverse domains for interdisciplinary applications. Both use cases are legitimate and an ideal terminology service should possess the capability to cater to both of these requirements.

2.2 Ontology development tools

Open-source ontology editors like Protégé and its online version WebProtégé [24] or visualization tools like WebVowl [25] can be used to inspect and develop ontologies one at a time. Yet when it comes to researching terms in multiple ontologies or comparing multiple ontologies at once, these tools lack the flexibility a TS provides. At the same time, they are tailored to experienced users and can be quite overwhelming for the novice user, due to the broadness and complexity of the functionalities they provide.

Such development tools are also not well suited for seeing changes between versions of the same ontology directly. To ensure applications will not break when implementing a new version of a reused ontology, it is however essential to enable ontology users to access and keep track of the changes between different ontology versions in a simple manner. This means all ontology versions should be explorable and comparable. If one wants to avoid loading different versions of the same ontology into multiple instances of one's ontology editor, one would usually resort to a TS. Yet, when it comes to providing multiple versions of multiple ontologies, the sheer amount of data and often missing versioning information can pose quite a challenge. Efforts in this direction on the pure ontological level have been made in the DBpedia Archivo project. Integrating version differences into a TS that also allows for browsing the terms of an ontology thus requires either high computational resources or new approaches around displaying changes, like the efforts behind the planned Knowledge Graph Change Language [26].

In addition to ontology archiving and versioning, there have been efforts to facilitate the ontology development and maintenance process. The main focus in facilitating ontology's development process was automation and quality assurance standards. For instance, ROBOT [27] was developed based on the standards of software development to help ontology developers automate common tasks like file conversions, error checking, reasoning, metadata annotation, modularization, and release management. The Ontology Development Kit (ODK) [28] integrates templates, standards, and quality checks with tools like ROBOT or GitHub in a bundled way using Docker to make their use in common development workflows easier.

As mentioned, ontology development tools are aimed at helping developers to have an optimal and precise development process. Also, they help the developer to have a better understanding of ontologies. However, they do not offer powerful ways to explore ontologies as much as TSs. On the other hand, TSs mainly focus on browsing and are not trying to provide further tools regarding the ontology development process. Considering these observations, an ideal TS should provide some tools to facilitate the development process to some extent, when it is used as a trusted main medium through which ontologies are perceived and understood.

3. Prolog - Finding the right words

A major challenge when using terminologies for data annotation or when importing parts of them in new terminologies is that one has to decide which terms to use from which terminology. In order to be able to make an appropriate decision, one must either already know which terminologies and terms are suitable for a specific scientific context and use case, or, more likely, one must be able to gather this knowledge by browsing the available terminologies. A terminology service with a graphical user interface (GUI) is such a resource, which renders terminologies comprehensible for humans, and should therefore be able to provide as much detailed information as possible on the terminologies themselves as well as on the terms they contain. This aspect is especially crucial when there is the need to compare multiple terminologies covering the same or overlapping knowledge domains. Terminology users need to be able to see the semantic differences and similarities as well as interdependencies between terminologies to make informed decisions. An intuitive way of browsing terminologies is thus key for evaluating the scope and use case applicability of terminology and for grasping its overall logical composition. It is important to be able to traverse the term hierarchy easily, similar to how we use file browsers or navigate publications in our document readers. In other cases, we need the ability to sort and filter the terms of terminology, like we are used to from working with spreadsheets. Such GUI functionalities are needed to keep focused when exploring the sometimes very complex semantics of terminology or to get to the desired information quickly in everyday workflows.

4. Ontology Development

To find out which features are required from a TS more concretely in collaborative ontology development workflows, we have analyzed the experiences made by domain and ontology experts in a still ongoing ontology development effort within the NFDI4Chem project [29]. The focus here was to benefit from a user-centered approach where we involve the ontology developers in our TS development process from the early stage to collect fine-grained user stories and requirements.

4.1 Vibrational Spectroscopy Ontology (VIBSO)

As the name implies, the intended domain covered by the Vibrational Spectroscopy Ontology is the discourse around a particular kind of spectroscopy that assays the vibrational modes of molecules and crystals. To be more precise, the ontology is meant to provide a formal representation of the technical terms used by domain experts to describe and share the research data output produced by such assays. It must thus contain terms that refer to the experimental setup, like the used devices, their relevant parts, and attributes, terms that refer to associated processes preceding and following the actual assay, like the preparation of the sample or the data transformation producing interpretable spectral images from raw signals, as well as terms that refer to relevant characteristics of these spectral images, like their dimensionality or analytical significance.

Adhering to best practices in ontology development [30], VIBSO's domain coverage depends heavily on reusing many terms from existing ontologies instead of defining them anew. First and foremost it depends on the Basic Formal Ontology [31] as a common ground for the abstract upper-level classes such as *material entity* or *process* and on the Relation Ontology [32] for commonly used relations. The Ontology for Biomedical Investigations [33] is another important and more concrete dependency of VIBSO, as it provides many general classes and some specific relations within the domain of scientific investigations, such as *assay*, *device,* or *protocol*. Most importantly, VIBSO's core depends on classes from the Chemical Methods Ontology [34], which already defines branches for the main chemical methods of interest - *vibrational spectroscopy* and *Raman spectroscopy* - as well as other general classes needed, such as *spectrum* or *spectrometer*. We are collaborating with the developers and maintainers of CHMO and due to the domain-specific overlap between it and VIBSO, there is the possibility to integrate VIBSO into CHMO in the future. At the moment, however, it seems best to keep the two separated to address the identified gaps and issues regarding VIBSO's scope.

4.1.1 Development Approach

Since most of the ontologies VIBSO depends on are part of the OBO Foundry [35] and to thus ensure interoperability, VIBSO's development also follows the best practices and principles [30] of this community. Furthermore, it is being developed in an iterative, version-controlled way that relies on continuous integration to make sure all changes are properly tracked and that the release files are quality controlled. For the technical implementation of this approach, we use the ODK, ROBOT, Protégé, GitHub, and the NFDI4Chem collection of our TS. The latter plays a central role in the development of VIBSO, as it is used to browse its most current version, to link to its terms in discussions, to search for requested terms in other domain-related ontologies, and to regularly look up the axiomatization patterns and term details of the reused ontologies. With regard to the conceptual aspect of our development approach, we rely on the collaboration with domain experts from chemistry and related scientific fields in which vibrational spectroscopies are being used, as they are the ones who know best what concepts are needed in this domain and how to label and define them. So far, we could benefit from the domain knowledge of scientists from the NFDI4Chem project, the CHARISMA project [36], BASF, and the Scuola Normale Superiore. These domain experts provided an initial list of terms from which a first ontology draft was created. Further term requests have since been filed and are being discussed with the domain experts mostly in regular open online calls and to a much lesser degree in the ontology's source code repository on GitHub.

As part of these development discussions, the domain experts are also asked to provide feedback on the used tools and workflows. This feedback is then used to find out how we can improve such a collaborative ontology development process more generally by enhancing the usability of the TS. The rationale behind this is not to turn the TS into yet another ontology editor that is just simpler to use. We also do not expect the domain experts to contribute to the source code of VIBSO directly, by having to learn the required specialized tooling. We rather want to develop simpler ways to communicate with them about the semantics of specific terms and terminologies. Making it easier in the TS to suggest new terms and changes to their metadata annotations (e.g. labels, definitions, or synonyms) or to discuss terminologies or terms in place, we believe that domain and ontology experts can benefit more from each other's expertise. We thus rather aim at making the TS a better tool to analyze and annotate the broad spectrum of available terminologies.

4.2 Lessons Learned from the VIBSO Development

Within the discussions between ontology engineers and domain experts, we have learned that tools to browse ontologies should be applicable in a simple and uncomplicated manner close to similar tools used by domain experts. Being still rather unfamiliar with the TS and its standard tree view for browsing an ontology, our domain experts requested to also have a tabular view that lists the available VIBSO terms to get a better overview. As a first step, we tried to address this need by switching from a pure Protégé based editing approach to one that uses spreadsheets and a TSV file as an input format for the definition of new VIBSO classes. Although this approach makes it easier for the domain experts to directly see, comment and edit the existing classes or create new ones by using common spreadsheet editors, it often seems to be too much of a hurdle to access and change this TSV via GitHub. In addition, the TSV defines only the classes from VIBSO and not the classes or relations imported from external ontologies. The domain experts thus have no direct way to understand the semantics of the latter. They would have to use the TS to look up why these terms are being used as parent classes or as part of VIBSO's axiomatization in the TSV. Being able to list all of the classes of an ontology in a tabular view directly in the TS avoids these downsides. If such a tabular view could also be used to sort and filter the class list, it would be easier than in the tree view to grasp or analyze certain details more intuitively. Sorting or filtering by term identifier would allow, for example, a more direct differentiation between imported and native terms, which allows one to quickly grasp the magnitude of external dependencies of an ontology. We believe that such a class list view could also gradually improve the acceptance and use of the tree view if the switching between the two views is implemented in a way that enables the user to learn to appreciate their differences in rendering an ontology.

When it comes to discussing term definitions and metadata or ontology-related questions as well as when one wants to request new terms, the most common approach in open-source development is to file issues in the Git-based repositories where the ontologies are maintained. Having to leave the TS to do so means an extra effort for the user. This has caused us to think about how we can use the TS as a medium to simplify such interactions. It would be more user-friendly to be able to at least list such external issues and in the best case read, write, or comment on them directly from within the TS. Like some of VIBSO's domain experts, ontology users who are unfamiliar with such external services would thus not need to learn a second platform to access more context information about an ontology, and to provide valuable feedback to its developers. At the same time also ontology users and developers who are familiar with the required Git workflows could benefit from this in their daily work by not having to switch contexts. Of course, such a feature should at best work with different version control platforms, like GitHub or GitLab. Fortunately, these two have suitable APIs and are very commonly used for open ontology development. So focusing on one of them can be considered already a great step forward.

Another aspect we have identified as a useful enhancement of the TS is the ability to add another layer of context information intended for a special user group. Apart from the metadata annotations provided on terms or the whole ontology, further documentation on design patterns and choices, which helps a user to better understand an ontology, is usually provided in scientific papers or discussions happening on mailing lists and development platforms. Well-curated ontologies contain links and notes to these sources. However, curating such links and notes is quite an effort, as the curators have to decide on their appropriateness to be included in the source code. Many use case or project-specific notes and discussions might thus be excluded or remain buried in the external sources. In addition, when following these links one usually has to leave the TS and thus runs the risk to get sidetracked. Especially when using multiple ontologies in a modular fashion, it can be quite challenging for less experienced users to understand why certain terms have to be imported, as they might seem unnecessary from a use case perspective but are needed from an ontological perspective to remain in line with their axiomatization. Having a functionality in the TS that allows its users to comment on the term and ontology level would be an alternative to sharing in-place insights.

As an example, with a note on the imported OBI term *assay* in VIBSO, which is an important superclass for the needed specializations of vibrational spectroscopy assays, we could provide directly further context to our domain experts about how to interpret the asserted and inferred axiomatization to remain in line with OBI design patterns, when defining new such specialized assays, instead of having to provide this information in the development documentation in other places. By making such a note and its particular context visible also on the same class in other ontologies, other TS users might benefit from such insights as well.

On the other hand, with notes on the ontology level, users can discuss their applicability in certain use cases. To remain in the VIBSO example, it would be quite helpful to add notes on CHMO that communicate to our domain experts the gaps and issues we have identified and need to address in VIBSO and link to their associated issues for further details. With such an additional layer, the CHMO developers as well as others could thus better keep track of our work without having to search multiple source code repositories for the related issues. For ontologies that are not maintained on an open platform that allows file issues, which unfortunately is still quite common, such a TS feature would be even more helpful. We believe that using a TS as such a medium could be a valuable way to keep a better overview of issues that span multiple ontologies and thereby better tackle complex issues of harmonization and mapping.

The experiences made by the ontology experts with the TS in the development of VIBSO and other NFDI4Chem-related ontology work, also lead to improvement suggestions for the TS. One major advantage of the TS for this user group is that for browsing and looking things up quickly in many different ontologies, they do not have to load all of them into Protégé. Another advantage of a TS is the ability to link to individual terms directly, which is an important way to reference these in many different contexts. Doing the lookup tasks in a web-based TS and the development tasks locally in Protégé helps a lot to keep focused, and keeping your focus is very important when working with formal ontologies. Yet, this user group is used to certain, often small but convenient, features that ease their daily workflows. Making such features also available in the TS where it makes sense is preferred. One of such features our ontology experts have identified as useful to port to the TS is the ability to traverse the hierarchy tree quickly with the keyboard navigation keys. Another one is the ability to adjust the size of GUI panes containing the tree view and term details. To find out which other features are also good candidates for porting to the TS, we will have to do further user research.

Table 1: Overview of lesson learned (LL) by observing the VIBSO development process.

Lesson Learned	Description
LL1	Extensive List View for classes.
LL2	Git issue list for ontologies.
LL3	Report issue and Term Request for ontologies.
LL4	Take/Read Notes on ontology/class/property/individual.
LL5	UI improvement such as keyboard navigation for tree views and resizing the tree view area pane.

5. Terminology Service

The development of our terminology service was initially motivated by the need to provide a robust service to search, browse, analyze, and access ontologies, terminologies, or vocabularies for various communities and scenarios like semantic annotation of research data or data generation. The TIB Terminology Service addresses these needs and provides overarching access to ontologies across multiple domains. Introducing the concept of *collections*, we can group ontologies by discipline, domain, or project providing customized views supporting communities not only to identify but also request new terminologies. The NFDI4Chem terminology service [37] is such a discipline-specific view for researchers interested in chemistry ontologies and has been used in VIBSO development. Table 2 summarizes available collections established so far.

Table 2: Collections used to bundle terminologies by domain or projects in the TIB Terminology Service.

Collections	Terminologies	Classes	Properties	Individuals
NFDI4Ing	53	483167	5840	3522
NFDI4Chem	38	147788	5928	26937
CoyPu	8	3767	2955	17289
NFDI4Culture	5	234	785	10
FID Move	9	121501	410	2439
FID BAU Digital	11	12820	1465	14702
FAIR Data Spaces	32	558036	4119	14068

Collections like NFDI4Chem are curated by community-agreed quality criteria [38], [39] applying workflows for suggesting new terminologies. Besides collections, ontologies can also be assigned to subjects like chemistry, physics, or engineering sciences. This classification can be applied to narrow down the ontologies of the TS to work with a specific set that is related to their scientific domain. The Terminology Service periodically checks updates for the indexed terminologies at the original sources to ensure it provides the latest version available.

5.1 Architecture

The TIB Terminology Service was designed following the Frontend-Backend pattern, which positions an ontology lookup service as the data-providing backend for a frontend application developed using the React library. The backend service is built upon the Ontology Lookup Service (OLS) developed by EBI. This architectural structure embodies a tightly integrated design with dedicated modules for ingesting and indexing terminologies. Additionally, it incorporates graph libraries that facilitate the visualization of these terminologies and also defines API methods that enable the presentation of data on the front-end. This architecture allowed us to take advantage of not only the pre-existing API but also the ontology ingestion process, which streamlined our development efforts. However, to ensure the architecture adequately served the unique requirements of the envisioned Terminology Service, we introduced new methods into the adopted system.

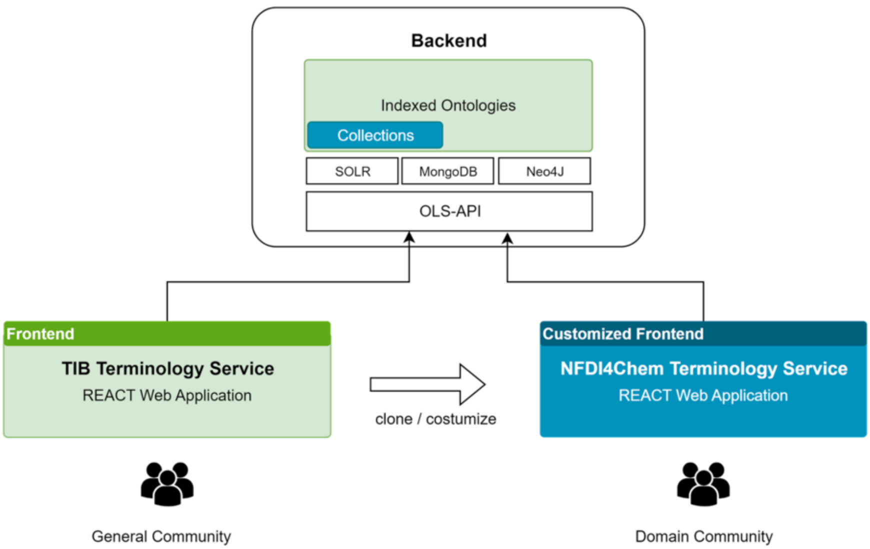


Figure 1: Architecture Terminology Service

The React frontend application addresses the need for flexible, autonomous implementation of features in the user interface, decoupled from the original approach of the OLS-web app.

As the original OLS user interface is embedded into the OLS-web app which is tightly integrated into the OLS backend architecture, serving individual communities would require running multiple OLS backend instances. This is costly for individual projects and hard to maintain.

We, therefore, decided for the TIB Terminology to ingest and index all ontologies in one backend while assigning the ontologies to one or more collections as we have described earlier. These collections are served to the respective communities by the front-end addressing their individual needs and preferences. Usually, collections can be accessed by choosing the filtered view in the search or browsing interface in the TIB TS. If required by a community the collections can be offered as highly customized, individual frontend applications. Such instances are cloned from the original source code of the TIB Terminology Service and are further tailored via configuration settings during the deployment process. This approach led to the development of customized solutions such as the NFDI4Chem Terminology Service. Such instances can not only be customized concerning the collection of terminology but also be enriched with customized components only available for the selected community. These components are centrally maintained in the TIB Terminology Service source code repository and can be activated during the deployment process of the customized front-end.

This architectural design ensures that the TIB Terminology Service is adaptable and responsive to the distinct needs of various user communities, providing them with customized interfaces and functionality while maintaining a centralized, coherent backend.

5.2 User-driven Implementation

The main motivation for implementation was to address the lesson learned and feedback that we observed through the VIBSO development process. As a result, we developed a new feature for each lesson learned in Table 1. Features under development can be accessed on the Terminology Service demo server (https://service.tib.eu/terminology). The source code is available via GitHub [40].

5.2.1 List of Github Issues for an Ontology

As highlighted in Section 4.2, an ideal Terminology Service (TS) should provide the necessary tools for facilitating the development and curation of high-quality, error-free ontologies. Moreover, many ontology developers leverage the GitHub version control system to optimize their development process, aligning it with established software development standards.

In light of this, enabling TS users to interact with GitHub directly through the TS emerged as an essential requirement. This interaction can take both passive and active forms. Passive interaction might involve tracking the discussions surrounding specific ontologies, while active interaction could entail participating in these discussions or even reporting issues.

As a result, we intended to provide a Terminology Service in such a way as to optimize the workflow for both the terminology developer and the end user. To achieve this, we developed a new feature that allows users to access the list of issues for a given ontology repository directly within the Terminology Service.

The feature aids users in tracking the progress of an ontology and being aware of the items in the ontology, which are being worked on. Moreover, it is useful to evaluate the quality of an ontology based on the latest activities, number, and kind of open issues. It can prompt both users and developers to take further steps. Finally, it reduces the mental load associated with context-switching, as users do not have to leave the TS to get to the GitHub issues. Currently, this feature is only available to the ontologies hosted and maintained on GitHub.

5.2.2 Issue Report and Term Request

In addition to monitoring existing repository issues for TS users, we recognized the importance of enabling them to report their issues, a need highlighted by our observations of VIBSO. This feature not only alleviates the need for a context switch between the TS and the target ontology repository (such as GitHub) but also empowers less experienced users of Git-based systems to actively participate in the development process by providing valuable feedback. Furthermore, it offers ontology developers the opportunity to enhance their work quality through an accessible and constructive feedback stream.

To address this need, we introduced a feature that allows TS users to create an issue on the target ontology repository hosted on GitHub. We added a button to the ontology homepage within the TS, which users can utilize to open an issue form, enter the issue content and title, and subsequently submit.

We categorized these issues into two groups: Generic Issues and Term Requests. The distinction was made because certain repositories, such as VIBSO, employ a specific template for filing new term requests. To uphold the integrity of this format, when a user selects the Term Request option, the TS automatically populates the input area with the appropriate template, thus informing users of its existence and encouraging its use.

5.2.3 Notes on Ontologies

As previously discussed, the capability of the Terminology Service (TS) to facilitate users in making or reviewing notes on ontologies was identified as an essential feature during the development of VIBSO. Such notes create an additional layer of context information atop the ontology or ontology terms, thereby benefiting both ontology developers and users. These notes serve as a dynamic feedback system that can either be used in cases where another tracker system is missing or where opening an issue is not desired. At the same time, they foster a deeper understanding of the ontology, by functioning as another avenue for documentation.

In pursuit of these advantages, we have developed a feature within TS that allows users to annotate ontologies with notes. To utilize this feature, a user navigates to the newly introduced "Note" tab in the ontology overview. Here, the user can add a new note, which requires a title, a body of text, and the Internationalized Resource Identifier (IRI) of the target artifact. The ability to specify target artifacts empowers users to apply notes to specific elements such as classes. At present, there are four possible target artifacts: Ontology, Class, Property, and Individual. Additionally, users can view a list of existing notes under the Note tab on the ontology page within TS.

Currently, this feature is in its prototype stage on our demo server. For its first stable release, we intend to enhance its usability by introducing more functionalities. For instance, we will enable users to filter the note list based on the target artifact. Furthermore, we will provide users with the ability to comment on a note about an ontology, thus opening up possibilities for discussion.

5.2.4 List View For Classes

One of the insights we gleaned was the necessity for the class view as a list presentation, in addition to the traditional tree view for ontology terms. We observed that the tree view could occasionally be perplexing for some novice users. Furthermore, certain operations, such as sorting based on the term ID, are not feasible within the tree view of classes.

As a response to these findings, we developed a paginated list view that exists alongside the tree view. This provides users with a comprehensive, tabular perspective of all the classes within an ontology. The list view is highly detailed, incorporating expansive metadata about terms, such as descriptions, relationships, and author comments.

Moreover, we have incorporated a 'jump-to' functionality within the list view. This allows users to quickly navigate to a specific term in the list by searching for its name, thereby enhancing the usability and efficiency of the platform.

5.2.5 UI enhancements

The final aspect of our implementation, based on the lessons learned, involved refining various UI-related features within our front-end application. The most common feedback pertained to the tree view page, where we present the term tree view alongside the term detail table when a term is selected.

The first concern addressed the challenge of navigating the tree solely using mouse clicks. To overcome this, we introduced keyboard arrow key functionality for navigation. Users can now use the right and left arrow keys to expand or collapse a node in the tree, while the up and down arrow keys allow movement between tree nodes.

The second enhancement dealt with the size of the detail table. Feedback suggested it was rather small for terms with extensive metadata. To improve this, we made the view pane resizable. This allows users to adjust the sizes of the tree view and detail table as needed, enhancing the readability of metadata and overall user experience.

6. Discussion

The topic of ontologies, with its abstract nature and philosophical undertones (ontology translates to 'doctrine of being'), often presents a challenging concept for domain experts to grasp. Understanding what terminologies or ontologies are, and why they are essential within their respective research areas, can be a complex process. The realization that a formal (i.e., not grounded in familiar classical/natural language) representation of knowledge (via ontologies) is necessary for machines to capture, comprehend, and interpret knowledge typically comes only after an extensive process of understanding. However, the importance and practicality of terminologies can be quite straightforward to demonstrate, especially in the context of the semantic description of research data through metadata annotation. In such a scenario, the Terminology Service (TS) can serve as an effective gateway tool, providing a swift and simple overview of the available ontologies. This tool can assist domain experts, for example, by providing domain-specific terminology collections that aid in the search and selection of suitable terms for their respective use cases. Further simplifying features, such as a list view, can provide even more support, enhancing the usability and efficiency of the service.

In this work, we employed a user-centered approach to identify the challenges and requirements in the design and development of a Terminology Service aimed at enhancing ontology development. We incorporated our target user group into the TS development process from its inception. In particular, we engaged in close contact with ontology development teams such as VIBSO, ensuring their perspectives and insights were woven into the fabric of the service from the earliest stages.

This collaboration allowed us to gain a deeper understanding of the ontology development process and the requisite features for an effective Terminology Service. We greatly valued the continuous feedback received from the ontology engineers, which proved crucial in shaping our service. Additionally, we endeavored to incorporate data annotators, who are primary users of the ontology, into our development process, thereby ensuring we cater to their needs effectively.

We facilitated numerous events and meetings, including conferences, workshops, and online sessions, with two primary objectives. Firstly, we sought to introduce our TS and demonstrate how it could be utilized for data annotation. Secondly, we aimed to gather feedback and user requirements to improve the quality of the ontologies and their presentation within the TS.

These close interactions with our users were instrumental in helping us conceptualize the ideal TS and identify its essential features. The insights gleaned from these engagements were directly translated into the design and functionality of our service, ensuring it adequately meets user needs and expectations.

Facilitating FAIR data annotation for researchers necessitates the provision of not just any ontologies, but those of high quality. Therefore, an ideal Terminology Service must cater to the needs of ontology developers, thereby enhancing the development process and ensuring quality. Our central finding was the necessity of integrating TS into the entire development process, from inception to conclusion. This implies the TS should address developer needs at various stages, such as artifact browsing and quality check, through discussion, feedback, and notes.

The ability of TS to offer these features offers multiple benefits to ontology developers. Firstly, it facilitates continuous feedback via Git issues and TS notes, ensuring the quality of developed ontologies. Furthermore, having notes and Git issues allows other TS users and less experienced developers to stay current with the latest discussions and topics concerning ontologies. This adds a new layer of context around the target ontology, enhancing its comprehensibility. Moreover, incorporating these ontology development tools into TS reduces context switching between different systems and environments, which could negatively impact a developer's performance.

Finally, we noted that the usability of the TS user interface (UI) is critical for ontology developers. There's a substantial demand for presenting a list view of classes alongside a tree view for an ontology. Given the high volume of terms in many ontologies, it is crucial for developers to swiftly navigate through the tree view using keyboard arrow keys. Additionally, due to the challenge of reading extensive metadata for a term in limited screen space, it is essential to allow TS users to resize different information box panes, thereby improving readability.

7. Conclusion

We have presented a user-centered development of our Terminology Service in the realm of research data annotation, where we actively engaged ontology developers, knowledge engineers, domain experts, and data annotators in the development process from its inception. Their continuous feedback and insights during its usage in an ongoing ontology development process proved instrumental in identifying requirements for the TS.

The added value of the TS by integrating it throughout the ontology development process is a central finding. The inclusion of features such as GitHub issue tracking, term requests, and ontology notes within the TS streamlines the development workflow, reduces context switching, and promotes collaboration among ontology developers and users.

Usability improvements have been another key focus. The introduction of an enhanced list view alongside the traditional tree view provides users with a comprehensive overview of ontology classes, enhancing efficiency and ease of navigation. The incorporation of keyboard arrow functionality and resizable information panes further improves the user interface, allowing for more intuitive interaction.

In conclusion, this work demonstrates the value of the TS for ontology developers and data annotators and proves its potential as an effective tool for ontology design and curation.

The next steps will include enhancing GitHub interaction features and the TS Note system to incorporate ontology curation into the terminology service user interface to establish the TS as an integrated ontology curation tool. We will continue the user-driven development to further support curation workflows from the very beginning. As ontologies continue to play a vital role in knowledge representation and data interoperability, the development and refinement of effective Terminology Services will contribute to advancing scientific understanding and collaboration across diverse domains.

Acknowledgements

We thank all PIs and developers at TIB who contribute to the TIB Terminology Service.

The presented work was conducted as part of the NFDI4Chem project (DFG project no. 441958208). The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding and support.

References

[1] "Re3Data," *Registry of Research Data Repositories*. [Online]. Available: https://www.re3data.org. [Accessed: May 30, 2023]

[2] S.-A. Sansone *et al.*, "FAIRsharing as a community approach to standards, repositories and policies," *Nat. Biotechnol.*, vol. 37, no. 4, pp. 358–367, Apr. 2019, doi: 10.1038/s41587-019-0080-8. [Online]. Available: http://dx.doi.org/10.1038/s41587-019-0080-8

[3] "EOSC Portal," *EOSC Portal*, Apr. 14, 2023. [Online]. Available: https://eosc-portal.eu/. [Accessed: May 29, 2023]

[4] N. Hartl, E. Wössner, and Y. Sure-Vetter, "Nationale Forschungsdateninfrastruktur (NFDI)," *Informatik Spektrum*, vol. 44, no. 5, pp. 370–373, Oct. 2021, doi: 10.1007/s00287-021-01392-6. [Online]. Available: https://doi.org/10.1007/s00287-021-01392-6

[5] S. Kanza, C. Willoughby, N. J. Knight, C. L. Bird, J. G. Frey, and S. J. Coles, "Digital research environments: a requirements analysis," *Digital Discovery*, Apr. 2023, doi: 10.1039/D2DD00121G. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2023/DD/D2DD00121G. [Accessed: May 29, 2023]

[6] M. D. Wilkinson *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship,"

Sci Data, vol. 3, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18. [Online]. Available: http://dx.doi.org/10.1038/sdata.2016.18
[7] M. Martone and Others, "Data citation synthesis group: Joint declaration of data citation principles," *San Diego CA: Force11. https://www. force11.org/group/joint-declaration-datacitafion-principles-final*, doi: 10.25490/a97f-egyk. [Online]. Available: http://dx.doi.org/10.25490/a97f-egyk
[8] "DataCite Schema," *DataCite Schema*. [Online]. Available: http://schema.datacite.org/. [Accessed: May 30, 2023]
[9] "DCMI Metadata Terms," *Dublin Core*. [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/. [Accessed: May 29, 2023]
[10] S. Kunis, S. Hänsch, C. Schmidt, F. Wong, C. Strambio-De-Castillia, and S. Weidtkamp-Peters, "MDEmic: a metadata annotation tool to facilitate management of FAIR image data in the bioimaging community," *Nat. Methods*, vol. 18, no. 12, pp. 1416–1417, Dec. 2021, doi: 10.1038/s41592-021-01288-z. [Online]. Available: http://dx.doi.org/10.1038/s41592-021-01288-z
[11] S. Jupp, T. Burdett, C. Leroy, and H. Parkinson, "A new Ontology Lookup Service at EMBL-EBI," *Workshop on Semantic Web Applications and Tools for Life Sciences*, 2015 [Online]. Available: https://www.semanticscholar.org/paper/b83bfbfc1f2f08e5b88af5ef65ef2a8687ac4112. [Accessed: May 11, 2023]
[12] M. A. Musen *et al.*, "The center for expanded data annotation and retrieval," *J. Am. Med. Inform. Assoc.*, vol. 22, no. 6, pp. 1148–1152, Nov. 2015, doi: 10.1093/jamia/ocv048. [Online]. Available: http://dx.doi.org/10.1093/jamia/ocv048
[13] N. Noy *et al.*, "BioPortal: A Web Repository for Biomedical Ontologies and Data Resources," *International Workshop on the Semantic Web*, 2008 [Online]. Available: https://www.semanticscholar.org/paper/cf2202e842500442eaea4c218425c98d599ee481. [Accessed: May 11, 2023]
[14] J. Pleiss, "Standardized data, scalable documentation, sustainable storage – EnzymeML as A basis for FAIR data management in biocatalysis," *ChemCatChem*, vol. 13, no. 18, pp. 3909–3913, Sep. 2021, doi: 10.1002/cctc.202100822. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/cctc.202100822
[15] *dataverse-external-vocab-support: Scripts and material related to using external vocabulary services with Dataverse*. Github [Online]. Available: https://github.com/gdcc/dataverse-external-vocab-support. [Accessed: May 29, 2023]
[16] P. Tremouilhac *et al.*, "Chemotion ELN: an Open Source electronic lab notebook for chemists in academia," *J. Cheminform.*, vol. 9, no. 1, p. 54, Sep. 2017, doi: 10.1186/s13321-017-0240-0. [Online]. Available: http://dx.doi.org/10.1186/s13321-017-0240-0
[17] A. Ledl, "Das BAsel Register of Thesauri, Ontologies & Classifications (BARTOC)," presented at the 104. Deutscher Bibliothekartag, Nürnberg, 2015 [Online]. Available: https://opus4.kobv.de/opus4-bib-info/files/1677/Poster_BARTOC_DBT_2015.pdf. [Accessed: May 11, 2023]
[18] J. Voß, J. M. Agne, U. Balakrishnan, and M. Akter, "Terminology Registries and Services," Nov. 2016, doi: 10.5281/zenodo.166717 [Online]. Available: https://zenodo.org/record/166717
[19] S.-Y. Yang, "OntoPortal: An ontology-supported portal architecture with linguistically enhanced and focused crawler technologies," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 10148–10157, Aug. 2009, doi: 10.1016/j.eswa.2009.01.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417409000402
[20] S. Jupp, T. Burdett, C. Leroy, and H. E. Parkinson, "A new Ontology Lookup Service at EMBL-EBI," in *SWAT4LS*, 2015.
[21] "SKOS Simple Knowledge Organization System Reference." [Online]. Available: https://www.w3.org/TR/skos-reference/. [Accessed: May 11, 2023]
[22] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, and N. Arndt, "DBpedia Archivo: A Web-Scale Interface for Ontology Archiving Under Consumer-Oriented Aspects," in *Semantic Systems. In the Era of Knowledge Graphs*, 2020, pp. 19–35, doi: 10.1007/978-3-030-59833-4_2 [Online]. Available: http://dx.doi.org/10.1007/978-3-030-59833-4_2
[23] P. Y. Vandenbussche and G. A. Atemezing, "Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web," *Semant. Pragmat.*, 2017 [Online]. Available: https://content.iospress.com/articles/semantic-web/sw213
[24] M. A. Musen and Protégé Team, "The Protégé Project: A Look Back and a Look Forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015, doi: 10.1145/2757001.2757003. [Online]. Available: http://dx.doi.org/10.1145/2757001.2757003
[25] V. Wiens, S. Lohmann, and S. Auer, "WebVOWL Editor: Device-Independent Visual Ontology Modeling," in *International Semantic Web Conference (P&D/Industry/BlueSky)*, 2018 [Online]. Available: http://ceur-ws.org/Vol-2180/paper-75.pdf
[26] *kgcl: Datamodel for KGCL (Knowledge Graph Change Language)*. Github [Online]. Available:

https://github.com/INCATools/kgcl. [Accessed: May 30, 2023]
[27] R. C. Jackson, J. P. Balhoff, E. Douglass, N. L. Harris, C. J. Mungall, and J. A. Overton, "ROBOT: A Tool for Automating Ontology Workflows," *BMC Bioinformatics*, vol. 20, no. 1, p. 407, Jul. 2019, doi: 10.1186/s12859-019-3002-3. [Online]. Available: http://dx.doi.org/10.1186/s12859-019-3002-3
[28] N. Matentzoglu *et al.*, "Ontology Development Kit: a toolkit for building, maintaining and standardizing biomedical ontologies," *Database* , vol. 2022, Oct. 2022, doi: 10.1093/database/baac087. [Online]. Available: http://dx.doi.org/10.1093/database/baac087
[29] C. Steinbeck *et al.*, "NFDI4Chem - Towards a National Research Data Infrastructure for Chemistry in Germany," *Research Ideas and Outcomes*, vol. 6, p. e55852, Jun. 2020, doi: 10.3897/rio.6.e55852. [Online]. Available: https://riojournal.com/article/55852/
[30] R. C. Jackson *et al.*, "OBO Foundry in 2021: Operationalizing Open Data Principles to Evaluate Ontologies," *bioRxiv*, p. 2021.06.01.446587, Jun. 02, 2021 [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.06.01.446587v1. [Accessed: Sep. 08, 2021]
[31] R. Arp, B. Smith, and D. A. Spear, *Building Ontologies with Basic Formal Ontology*. Cambride, London: MIT Press, 2015.
[32] B. Smith *et al.*, "Relations in biomedical ontologies," *Genome Biol.*, vol. 6, no. 5, p. R46, Apr. 2005, doi: 10.1186/gb-2005-6-5-r46. [Online]. Available: http://dx.doi.org/10.1186/gb-2005-6-5-r46
[33] A. Bandrowski *et al.*, "The Ontology for Biomedical Investigations," *PLoS One*, vol. 11, no. 4, p. e0154556, Apr. 2016, doi: 10.1371/journal.pone.0154556. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0154556
[34] R. S. C. Ontologies, *rsc-ontologies - Overview*. Github [Online]. Available: https://github.com/rsc-ontologies. [Accessed: May 24, 2023]
[35] B. Smith *et al.*, "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration," *Nat. Biotechnol.*, vol. 25, no. 11, pp. 1251–1255, Nov. 2007, doi: 10.1038/nbt1346. [Online]. Available: http://dx.doi.org/10.1038/nbt1346
[36] "H2020 CHARISMA," *H2020 CHARISMA*. [Online]. Available: https://www.h2020charisma.eu/. [Accessed: May 29, 2023]
[37] "NFDI4Chem Terminology Service," *NFDI4Chem Terminology Service*. [Online]. Available: https://terminology.nfdi4chem.de. [Accessed: May 31, 2023]
[38] P. Strömert, J. Hunold, A. Castro, S. Neumann, and O. Koepler, "Ontologies4Chem: the landscape of ontologies in chemistry," *Pure Appl. Chem.*, Mar. 2022, doi: 10.1515/pac-2021-2007. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/pac-2021-2007/html. [Accessed: May 02, 2022]
[39] P. Strömert, J. Hunold, and O. Koepler, "1st Ontologies4Chem Workshop – Ontologies for chemistry," Sep. 07, 2022. [Online]. Available: https://www.confident-conference.org/index.php/Event:13b31d0e-f398-4bc8-a8c2-54352b1dcc66. [Accessed: Sep. 28, 2022]
[40] *TIB-Terminology-Service-Frontend-2.0: Frontend source code for NFDI4Chem, NFDI4Ing, TIB TS v2.0*. Github [Online]. Available: https://github.com/TIBHannover/TIB-Terminology-Service-Frontend-2.0. [Accessed: May 31, 2023]

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230006

Perplexed by Idioms?

J. Nathanael PHILIPP [a,b,1], Max KÖLBL [c], Erik DAAS [b], Yuki KYOGOKU [b] and Michael RICHTER [b]

[a] *Serbski institut*
[b] *Leipzig University*
[c] *Osaka University*

ORCiD ID: J. Nathanael Philipp https://orcid.org/0000-0003-0577-7831, Max Kölbl https://orcid.org/0000-0001-7460-4139, Erik Daas https://orcid.org/0009-0007-0879-0878, Michael Richter https://orcid.org/0000-0001-7460-4139

Abstract. The aim of this study is to identify idiomatic expressions in English using the measure *perplexity*. The assumption is that idiomatic expressions cause higher perplexity than literal expressions given a reference text. Perplexity in our study is calculated based on n-grams of (i) PoS tags, (ii) tokens, and (iii) thematic roles within the boundaries of a sentence. In the setting of our study, we observed that no perplexity in the contexts of (i), (ii) and (iii) manages to distinguish idiomatic expressions from literals. We postulate that larger, extra-sentential contexts should be used for the determination of perplexity. In addition, the number of thematic roles in (iii) should be reduced to a smaller number of basic roles in order to avaiod an uniform distribution of n-grams.

Keywords. perplexity, idiomatic expressions, literal expressions, information theory

1. Introduction

Idiomatic expressions, or phraseologies, (in the following **IE**) such as the verb-noun compounds (**VNC**) *kick the bucket*, *hit the fan*, *blow whistle*, *hit the sack* or *lose face* are ubiquitous in the English language. The aim of this study is to automatically identify IE in English texts using an information theoretic framework [1].

IE are far less subject to the principle of compositionality than literal expressions [2,3] since in most cases, the meaning of an expression cannot straightforwardly be derived from the meaning of its parts. The interpretation of idioms thus is not reductionist. So, to unlock the meaning of the IE for instance in the sentence *the shit has hit the fan at our house* it is not sufficient to know the meaning of *shit*, *hit* and *fan* alone. Understanding an IE touches on conventionality in language, since its meaning has evolved through specific language usage and convention. [2] emphasise that *[t]he meaning of IE involves i.a. metaphors and hyperboles*, and the meaning of the constituents of IE is overruled. IE are stable linguistic constructions, mostly with specific syntax as in *lose face* or *blow whistle*, a feature referred to as *(In)flexibility* [2,3]. This feature also means the imper-

[1]Corresponding Author: J. Nathanael Philipp, nathanael@philipp.land.

meability of IE, i.e., grammatical transformations, extractions and insertions lead to ungrammaticality, as the sentences in (1) exemplify. ('*' indicates ungrammaticality):

(1)a * The fan at our house was hit by the shit. (passive transformation: the object becomes the grammatical subject)
b * It was the fan that the shit has hit. (clefting)
c * The shit has hit the wanky fan at our house. (insertion)

In this study, we investigate whether it is possible to distinguish IE from literal expressions by the divergence measure of *perplexity* which is close related to the information theoretic feature of *surprisal* [4,5,6,7]. So, we ask whether IE cause higher perplexity than literals given a reference data set. The amount of perplexity which is exactly defined in the Eqs. 1 and 2, represents the amount of surprisal which is contextualised information and is to be interpreted as the deviation from the expected. The higher the surprisal of a linguistic unit, and thus the higher the perplexity it causes, the greater the cognitive effort is to process it [5,7] which is an empirically proven claim (see amongst others [8,9]).

In our study, we measure the divergence of IE and literals from reference texts, and the assumption is that these divergences should differ. In the data set of the study (see Section 3), the VNC can be understood both literally or as IE.

We determine perplexity as the divergence between n-gram-probabilities of linguistic patterns in two sorts of data sets on the one hand, i.e., one that contains IE and another set that does not, and on the other hand two large reference data sets.[2]

What is the practical interest in dealing with the identification of IE? While some state-of-the-art translation programs and chat bots are able to recognise IE and provide correct translations and interpretations, some others do not. Google Translator for instance translates *yesterday the shit has hit the fan at our house* into German as *gestern hat die Scheiße den Lüfter unseres Hauses getroffen* which is literally correct, but does not make any sense at all. DeepL and in particular OpenAI/ChatGPT do a much better job in this respect. However, the latter shows slight weaknesses in interpreting the German IE *nicht auf den Kopf gefallen sein* which means to be clever and smart. We had a dialogue in with ChatGPT from Open AI[3].

Question: Wir sind nicht auf den Kopf gefallen. Was heißt das? lit: 'We did not fall onto our heads. What does that mean?'

Answer: Dieser Ausdruck bedeutet, dass man klug und clever ist und nicht dumm. Man ist in der Lage, kluge Entscheidungen zu treffen und ist nicht leicht hinters Licht zu führen. 'This expression means that one is smart and clever and not stupid. One is able to make wise decisions and is not easily fooled.'

This is a really good answer, the chat bot recognises the IE and explains it well. Now consider the following dialogue:

Question: Wir sind nicht vom Fahrrad auf den Kopf gefallen. Was heißt das? 'we did not fall off our bikes onto our heads. What does that mean?'

[2]Doing so, we are inspired by the workflow in [10] on classification of translations by perplexity.
[3]https://chat.openai.com

Answer: *Diese Redewendung bedeutet, dass man nicht dumm oder unerfahren ist, sondern dass man gut informiert und handlungsfähig ist. Es drückt auch die Zuversicht aus, dass man in einer bestimmten Situation erfolgreich sein wird.* 'This idiom means that one is not stupid or inexperienced, but that one is well informed and able to act. It also expresses confidence that one will succeed in a given situation.'

This explanation however is not correct, since *vom Fahrrad* is inserted into IE whereby the expression lost its idiomatic character and is to be understood literally. ChatGPT/OpenAI however interpreted also this sentence as an idiomatic expression.

In the present study, the identification of IE is unsupervised and as a first step will be based on n-grams of PoS-tags [10] which yields perplexity expressing grammatical surprisal. Secondly, we will consider n-grams of tokens as a representation of the respective data sets which yields perplexity expressing lexical surprisal. Thirdly, we will consider sequences of *thematic roles*, i.e., semantic types of entities, processes and events as defined by [11,12] which yield perplexity expressing semantic surprisal.[4] In the due course of this paper, we will use the latter term. [15, p. 48] describes thematic roles as follows:

Thematic roles are generalisations among the arguments of a predicate in order to capture regularities between the semantic representation and the syntactic expression of that predicate.

Very elementary thematic roles are, for example, *agent* and *patient*, i.e., the participants in transitive scenes in the world and transitive linguistic constructions that represent those scenes. A sentence like *we eat a chocolate* could be assigned the thematic roles **Agent-Process-Patient**.

2. Previous work on automatic detection of idiomatic expressions

To the best of our knowledge there is no work on identification of idioms within the framework of information theory. A recent study on the automatic classification of phraseologies by [16] reports a unsupervised, classification of IE based on topic detection. The authors assume that words that are highly relevant in the main topic of discourses are not very likely to occur in IE. In other words, IE are assumed to be semantically distinct from the main topic of the discourse, and, in addition, the study brings to light that IE are associated with a higher level of affectivity. This was proposed already in an earlier study [17], in which the authors state that IE are semantic 'outliers' in a given context and thus cause surprisal. Identification of IE in [17] is carried out by the *Principal Component Analysis*.

The model in [18] first generates both static and contextualised word embeddings. Additional information such as PoS-tags is incorporated in the attention phase, and in the enriched static phase, embeddings are further combined with the contextualised embeddings. This is input to a BiLSTM-neural network: if the contextualised representation of a word is semantically compatible with its context, it is literal; if not, it is figurative.

[19] propose a model that is characterised by syntagmatic and context features, and, in addition, by other features such as the number of words in a collocation. For the syntagmatic feature, both count-based and predictive models are proposed (for the two

[4] [13] notes that frame semantics has two origins: a linguistic origin from Fillmore's case grammar and an origin from 'Artificial Intelligence' in [14].

models, see [20]). The effect of each feature varies according to the characteristics of the datasets.

[19] conclude that the context feature contributes to detecting semantically dissimilar words, while the count-based measure contributes to assessing the fixedness of collocations.

3. Dataset, concepts and technique of analysis

The data on which this study is based have been extracted from the *British National Corpus* (BNC)[5] and were already used in [16][6]. The dataset comprises 1997 sentences with idioms and 535 literal expressions. To avoid any bias, we split the data into five data sets of almost even size: four sets with about 500 sentences each which contain idioms and one set with 535 sentences containing literal expressions. As in [16], only verb-noun constructions (VNC) are in the focus of the present study. For the labeling of VNC as IE or literal expression, [16] used the list in [21,22]. [16] treated idiomacity as a binary property and explicitly not as a gradual property [23].

In order to determine the perplexity of IE and literals, we use two types of reference data sets, namely a news corpus with 1M sentences from the *Wortschatz Leipzig* (eng_news_2020_1M) corpora collection[7], and a Wikipedia corpus also with 1M sentences, taken from the same source (eng_wikipedia_2016_1M).

For PoS-tagging, we employ the *spaCy*[8] parser which assigns 15 PoS tags. Thematic roles are assigned by *LOME*, a *system for multilingual information extraction* [24]. For entity-assigns makes use of about 2000 thematic roles[9]. Specifically, we used for the entity-type parsing the program *span-finder*[10].

3.1. Perplexity: Measure of surprisal

Perplexity (PP) is a measure of how well a probability distribution in a statistical language model predicts a data sample[11]. It is defined as an two to the power of the entropy H of a probability distribution as exponent, as given in Eq. 1; the lower the perplexity, the better the model.

$$PP = 2^{H} \tag{1}$$

In this study, H is the *conditional entropy*: expressed in terms of information theory, the conditional entropy is a measure of the quality of a model for a probability distribution q, given a true distribution p. This is reminiscent of the idea behind the Kullback-Leibler divergence, and like this, conditional entropy is not symmetrical. PP uses condi-

[5] https://github.com/bondfeld/BNC_idioms
[6] The data were made available by Jing Peng and Anna Feldman.
[7] https://wortschatz.uni-leipzig.de/de
[8] https://spacy.io
[9] https://framenet.icsi.berkeley.edu/fndrupal/frameIndex
[10] https://github.com/hiaoxui/span-finder
[11] https://en.wikipedia.org/wiki/Perplexity

tional entropy to indicate the degree perplexity, when a model's prediction is compared with a data sample. For the calculation of entropy, we use Eq. 2:

$$H = -\sum_{w,c} q(w,c) \log_2 p(w|c) \tag{2}$$

w denotes a linguistic unit and c its context. For the PoS-tags, each dataset is represented as a combined distribution of bi- to heptagrams, for the tokens, each dataset is represented as a combined distribution of the bi- and trigrams, and for the thematic roles a combined distribution of the bi- to tetragrams is used. PP expresses the divergences between these probabilities.

4. Results

Table 1 gives the perplexity between the reference data sets and the BNC data, i.e., four data sets with IE (*Idioms* 1–4) and one data set with exclusively literal expressions (*Literals*).

Reference dataset	**Idioms 1**	**Idioms 2**	**Idioms 3**	**Idioms 4**	**Literals**
Wikipedia	6.06	6.11	6.11	6.18	6.06
News	5.92	6.00	6.00	6.03	5.93

Table 1. Perplexity based on PoS-tag probabilities between reference data sets, literals and idioms.

Table 1 shows approximately the same perplexity for all IE and literals: for n-grams and POS tags, IE and literals are each distributed similarly to the reference data.**Table** 2 gives the perplexity between the reference data and the BNC data based on bi- and trigrams of tokens.

Reference dataset	**Idioms 1**	**Idioms 2**	**Idioms 3**	**Idioms 4**	**Literals**
Wikipedia	19.30	18.63	18.63	17.74	18.27
News	19.54	18.66	18.66	18.19	18.52

Table 2. Perplexity based on bi- and trigram probabilities of tokens between reference data sets, literals and idioms.

Again, there is hardly any difference in the perplexity caused by idioms and literals. We observe that the perplexity values are much higher here than for PoS-tags which is due to the fact that the set of PoS-tags is considerably smaller than that of the tokens. Recall Eq. 1: high average information due to low probabilities of signs causes high perplexity and vice versa. N-grams of tokens are more informative than n-grams of PoS tags. **Table** 3 shows the perplexity of semantic surprisal from thematic roles.

Here, the same picture emerges as in the **Tables** 1 and 2: based on semantic surprisal of thematic roles, literals and idioms cannot be distinguished from each other by perplexity. We notice again that the amount of perplexity correlates with the number of linguistic units it is based on. The results in **Table** 3 are derived from (about) 2000 thematic roles, and thus the values are higher than those based on 15 PoS tags, but they are lower than the values derived from the large, entire set of tokens.

Reference dataset	Idioms 1	Idioms 2	Idioms 3	Idioms 4	Literals
Wikipedia	14.49	14.49	14.49	14.09	14.40
News	14.03	13.97	13.97	13.34	13.71

Table 3. Perplexity based on bi- and trigram probabilities of thematic roles between reference data sets, literals and idioms.

5. Discussion and Conclusion

In our study, IE could not be identified by any type of perplexity. The observation that perplexity from PoS tags as grammatical surprisal fails is plausible because all IE in the corpus are verb-noun compounds (VNC). They structurally correspond to the VNC of the literals. We can interpret perplexity of grammatical surprisal as a baseline, and moreover, this shows that perplexity is a suitable measure for our aims: when perplexity is based on grammatical structures, VNC in IE and literals should not exhibit large differences. This is what we observe. However, there is also no perplexity with lexical surprisal, and this is not what we expected. That is, n-grams of tokens do not give the language processor any clue to distinguish IE from literals. This is actually a semantic distinction. This finding is surprising because in the common and influential distributional language model [25], it is assumed that the meaning of a linguistic unit is represented in its context. Prototypically, contexts in this model mean co-occurrences of a linguistic unit in a pre-defined context window. Similar contexts indicate a similar meaning of a linguistic unit, while different contexts indicate a different meaning. We attribute the problem with the n-grams of tokens in our study to the small context sizes. Contexts of bi- to tetragrams cannot represent a larger conversational context, let alone a discourse. So, for example, if the word *ganache*, i.e., a soft chocolate filling, occurs in the conversational context of, say, quantum mechanics, it is presumably not possible to derive from a bigram-context the high surprisal / perplexity that *ganache* in this special context causes. The same effect emerges with semantic surprisal, since n-grams of thematic roles within the sentence boundaries do not yield higher perplexity with IE than with literals.

The restriction to local contexts within the sentence boundaries is due to our data basis, because literals and idioms are represented as isolated sentences here. However, this is a crucial shortcoming, since Levy's definition of surprisal [7] provides for extra-sentential contexts. According to this definition, surprisal is the deviation from the expected based on co-occurrence, but also from a larger context which can be the discourse, an entire corpus, or a paragraph, in short, any linguistic material that exceeds sentence boundaries. The perplexity scores that we calculated for lexical and semantic surprisal underpin Levy's definition, since we observed that inner-sentential contexts are not sufficient to distinguish IE from literals. A language processor needs thus a larger context to decide whether an expression can be understood literally or as IE. Derivation of surprisal / perplexity from extra-sentential contexts is thus a task of future work.

Another problem in calculating semantic surprisal concerns the large number of thematic roles that LOME assigns which makes the possibility quite high, that a given context will appear only once and finally, this would lead to uniform perplexity values. In future work, we will consider a reduction to 'basic' thematic roles, as initially introduced by Fillmore [11] such as *agent*, *patient*, *theme*, *instrument* and *source*.

References

[1] Shannon CE. A mathematical theory of communication. The Bell system technical journal. 1948;27(3):379-423.

[2] Espinal MT, Mateu J. Idioms and phraseology. In: Oxford Research Encyclopedia of Linguistics. Oxford University Press; 2019. .

[3] Nunberg G, Sag IA, Wasow T. Idioms. Language. 1994;70(3):491-538.

[4] Tribus M. Information theory as the basis for thermostatics and thermodynamics. Journal of Applied Mechanics. 1961 03;28(1):1-8. Available from: https://doi.org/10.1115/1.3640461.

[5] Hale J. A probabilistic Earley parser as a psycholinguistic model. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. Association for Computational Linguistics; 2001. p. 1-8.

[6] Hale J, Lutz D, Luh WM, Brennan J. Modeling fMRI time courses with linguistic structure at various grain sizes. In: Proceedings of the 6th workshop on cognitive modeling and computational linguistics; 2015. p. 89-97.

[7] Levy R. Expectation-based syntactic comprehension. Cognition. 2008;106(3):1126-77. Available from: https://doi.org/10.1016/j.cognition.2007.05.006.

[8] McCusker L. Some determinants of word recognition: Frequency. In: 24th annual convention of the southwestern psychological association, fort worth, tx; 1977. .

[9] Smith NJ, Levy R. The effect of word predictability on reading time is logarithmic. Cognition. 2013;128(3):302-19.

[10] Bizzoni Y, Lapshinova-Koltunski E. Measuring Translationese across Levels of Expertise: Are Professionals more Surprising than Students? In: NoDaLiDa; 2021. p. 53-63.

[11] Fillmore CJ. l968. The case for case." by E. Bach and R. Harms, l-88. New York: Holt, Rinehart & Winston, Inc; 1968.

[12] Fillmore CJ, et al. Frame semantics and the nature of language. In: Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech. vol. 280. New York; 1976. p. 20-32.

[13] Hamm F. Frame semantics. The cambridge encyclopedia of the language sciences, Cambridge: CUP. 2009.

[14] Minsky M. A framework for representing knowledge. MIT, Cambridge; 1974.

[15] Butt M. Theories of Case. Cambridge Textbooks in Linguistics. Cambridge University Press; 2006.

[16] Peng J, Feldman A, Vylomova E. Classifying idiomatic and literal expressions using topic models and intensity of emotions. arXiv preprint arXiv:180209961. 2018.

[17] Feldman A, Peng J. Automatic detection of idiomatic clauses. In: Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I 14. Springer; 2013. p. 435-46.

[18] Zeng Z, Bhat S. Idiomatic expression identification using semantic compatibility. Transactions of the Association for Computational Linguistics. 2021;9:1546-62.

[19] Amin M, Fankhauser P, Kupietz M, Schneider R. Shallow context analysis for german idiom detection. Proceedings of the Shared Task on the Disambiguation of German Verbal Idioms at KONVENS 2021. 2021.

[20] Fankhauser P, Kupietz M. Count-Based and Predictive Language Models for Exploring DeReKo. In: Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-10); 2022. p. 27-31.

[21] Cook P, Fazly A, Stevenson S. The VNC-tokens dataset. In: Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008). Citeseer; 2008. p. 19-22.

[22] Fazly A, Cook P, Stevenson S. Unsupervised type and token identification of idiomatic expressions. Computational Linguistics. 2009;35(1):61-103.

[23] Pradhan M, Peng J, Feldman A, Wright B. Idioms: Humans or machines, its all about context. In: Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part I 18. Springer; 2018. p. 291-304.

[24] Xia P, Qin G, Vashishtha S, Chen Y, Chen T, May C, et al. LOME: Large ontology multilingual extraction. arXiv preprint arXiv:210112175. 2021.

[25] Rubenstein H, Goodenough JB. Contextual correlates of synonymy. Communications of the ACM. 1965;8(10):627-33.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230007

Native Execution of GraphQL Queries over RDF Graphs Using Multi-Way Joins

Nikolaos KARALIS [a,1], Alexander BIGERL [a] and Axel-Cyrille NGONGA NGOMO [a]
[a] *DICE group, Department of Computer Science, Paderborn University*
ORCiD ID: Nikolaos Karalis https://orcid.org/0000-0002-0710-7180, Alexander Bigerl https://orcid.org/0000-0002-9617-1466, Axel-Cyrille Ngonga Ngomo https://orcid.org/0000-0001-7112-3516

Abstract. *Purpose:* The query language GraphQL has gained significant traction in recent years. In particular, it has recently gained the attention of the semantic web and graph database communities and is now often used as a means to query knowledge graphs. Most of the storage solutions that support GraphQL rely on a translation layer to map the said language to another query language that they support natively, for example SPARQL. *Methodology:* Our main innovation is a multi-way left-join algorithm inspired by worst-case optimal multi-way join algorithms. This novel algorithm enables the native execution of GraphQL queries over RDF knowledge graphs. We evaluate our approach in two settings using the LinGBM benchmark generator. *Findings:* The experimental results suggest that our solution outperforms the state-of-the-art graph storage solution for GraphQL with respect to both query runtimes and scalability. *Value:* Our solution is implemented in an open-sourced triple store, and is intended to advance the development of representation-agnostic storage solutions for knowledge graphs.

Keywords. graphql, knowledge graphs, multi-way joins

1. Introduction

Knowledge graphs serve as the data backbone of an increasing number of applications. Examples of such applications include search engines, recommendation systems, and question answering systems [1,2]. Consequently, efficient storage and querying solutions for knowledge graphs are imperative. Many triple stores [3,4,5,6,7,8,9,10] and graph databases [11,12] have hence been developed in recent decades. Used primarily by the semantic web community, triple stores process RDF knowledge graphs. A popular representation model among the graph database community is the property graph model [2,13]. While SPARQL is the designated query language for RDF, multiple languages have been developed to query property graphs (e.g., Cypher [14] and Gremlin [15]). Re-

[1]Corresponding Author: Nikolaos Karalis, nkaralis@mail.uni-paderborn.de. This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860801 and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070305. It has also been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D.

cently, GraphQL, a query language for APIs, has attracted the attention of both the graph database [16,17] and the semantic web [18,19,20,21] communities.

The focus of the semantic web community regarding GraphQL has been on the development of query translation tools [18,20,21]. These tools translate GraphQL queries into SPARQL queries, which are then issued to a triple store. A drawback of such solutions is that the results produced by triple stores need to be rewritten, since GraphQL dictates a strict response format. This process adds a significant overhead to the execution of queries [18, Table 3], especially in cases where the results are large. To the best of our knowledge, there are no publicly available triple stores that treat GraphQL as a first-class citizen.

While most constructs for processing basic graph patterns can be exploited in a straightforward manner for GraphQL processing, the formal semantics of GraphQL [22] demand the use of left-join operations for the evaluation of GraphQL queries. However, conventional two-way left-join operations are not suitable for the evaluation of GraphQL queries, as the results of such queries can be constructed incrementally [22]. We hence focus on presenting a *novel multi-way left-join algorithm* inspired by worst-case optimal join algorithms [23], which can be used to enumerate GraphQL queries incrementally. By implementing our approach into a state-of-the-art triple store, we provide the *first publicly available triple store that treats GraphQL queries as first-class citizens*. We carried out an extensive evaluation using a synthetic benchmark generator, namely LinGBM [24], and the results suggest that our implementation is able to outperform a state-of-the-art graph storage solution providing GraphQL support, namely Neo4j.

The rest of this paper is structured as follows. The preliminaries are provided in Section 2. In Section 3, we present our multi-way left-join algorithm and show how to natively evaluate GraphQL queries over RDF graphs. We evaluate our approach in Section 4. We discuss related works in Section 5 and conclude in Section 6.

2. Preliminaries

Here, we introduce the concepts and the semantics of GraphQL that we use throughout this work along with their formal definitions as per [22]. We also briefly introduce worst-case optimal multi-way join algorithms, which have inspired our proposed algorithm.

2.1. GraphQL

GraphQL is a query language that was designed to simplify communication between clients and application servers. One of the main characteristics of GraphQL is that it is strongly typed. GraphQL services—i.e., servers and data sources whose data can be accessed and modified via GraphQL operations—expose a GraphQL schema to their clients by which incoming requests must abide. This schema defines a type system that describes the structure of the underlying data of the GraphQL service and the operations the service supports. Another important aspect of GraphQL is the hierarchical structure of its operations and responses. GraphQL operations form a tree structure that specifies the traversal on top of the underlying graph and the information that should be extracted from the nodes at each step of the traversal. In turn, the responses should follow the hierarchy defined by their respective operation. The syntax and capabilities of GraphQL, as well as the responsibilities of GraphQL services, are detailed in the language's official specification [25]. Even though the specification describes how services should handle the requests they receive, it does not provide a formal specification of the semantics of the language. Consequently, studying the expressiveness and complexity of the language remained a challenge. To tackle the lack of formal semantics and the consequences thereof, Hartig and Pérez [22] provide formal semantics for GraphQL queries that consist of *fields*, *field aliases* and *inline fragments*. The semantics rely on the formal definition of GraphQL schemata and graphs as well as the formalized syntax of GraphQL queries. Here, we reintroduce the definitions presented in [22].

The formal definitions presented below rely on the following sets. Let *Fields* be an infinite set of field names and $F \subset Fields$ a finite subset of *Fields*. Let A and T be finite sets of argument names and type names, respectively, where T is the disjoint union of O_T (object type names), I_T (interface type names), U_T (union type names) and *Scalars* (scalar type names). Last, let $L_T = \{[t] \mid t \in T\}$ be the set of list types constructed from T and *Vals* be a set of scalar values.

Definition 2.1 (GraphQL schema [22]) *A GraphQL schema $\mathcal{S}$ over (F, A, T) is composed of the following components:*

- $fields_{\mathcal{S}} : (O_T \cup I_T) \rightarrow 2^F$ *that assigns a set of fields to every object type and every interface type,*
- $args_{\mathcal{S}} : F \rightarrow 2^A$ *that assigns a set of arguments to every field,*
- $type_{\mathcal{S}} : F \cup A \rightarrow T \cup L_T$ *that assigns a type or a list type to every field and argument, where arguments are assigned scalar types,*
- $union_{\mathcal{S}} : U_T \rightarrow 2^{O_T}$ *that assigns a nonempty set of object types to every union type,*
- $impl_{\mathcal{S}} : I_T \rightarrow 2^{O_T}$ *that assigns a set of object types to every interface,*
- $root_{\mathcal{S}} \in O_T$ *that represents the* query *root type.*

Example 2.1 *Consider the following GraphQL schema $\mathcal{S}$*

```
interface Entity {
  id:String
  email:String
}
type Person impl Entity {
  id:String
  fname:String
  lname:String
  email:String
  age:Int
}
type Company impl Entity {
  id:String
  name:String
  email:String
  employees:[Person]
}
type Query {
  people(lname:String):[Person]
  companies:[Company]
}
schema { query:Query }
```

Let $F = \{$ `people, companies, employees, fname, age, id, lname, email, name` $\}$, $A = \{$ `lname` $\}$, $O_T = \{$ `Query, Company, Person` $\}$, $I_T = \{$ `Entity` $\}$, $U_T = \{\}$, *and Scalars* $= \{$ `String, Int` $\}$. *The GraphQL schema $\mathcal{S}$ is formalized over (F, A, T) as follows (we omit several assignments for brevity):*

$args_{\mathcal{S}}(\texttt{people}) = \{\texttt{lname}\}$, $fields_{\mathcal{S}}(\texttt{Entity}) = \{\texttt{id, email}\}$, $type_{\mathcal{S}}(\texttt{id}) = \texttt{String}$, $fields_{\mathcal{S}}(\texttt{Person}) = \{\texttt{id, fname, lname, email, age}\}$, $root_{\mathcal{S}} = \texttt{Query}$.

In practice, GraphQL services are implemented on top of data sources that adopt different data models (e.g., relational databases and graph databases). To provide the semantics of GraphQL queries in a unified manner, Hartig and Pérez [22] introduced the notion of GraphQL graphs. GraphQL graphs are logical constructs that abstract the underlying data sources of GraphQL services.

Definition 2.2 (GraphQL graph [22]) *A GraphQL graph over* (F,A,T) *is a tuple* $G = (N,E,\tau,\lambda,\rho)$ *with the following elements:*

- N *is a set of nodes,*
- E *is a set of edges of the form* $(u, f[a], v)$*, where* $u,v \in N$*,* $f \in F$*, and a is a partial mapping from A to Vals,*
- $\tau : N \to O_T$ *is a function that assigns a type to every node,*
- λ *is a partial function that assigns a scalar value* $\nu \in$ *Vals or a sequence* $[\nu_1 \dots \nu_n]$ *of scalar values* ($\nu_i \in$ *Vals*) *to some pairs of the form* $(u, f[a])$ *where* $u \in N$*,* $f \in F$ *and a is a partial function from A to Vals,*
- $\rho \in N$ *is a distinguished node called the root node.*

Definition 2.3 (GraphQL query [22]) *A GraphQL query over* (F,A,T) *is an expression* ϕ *constructed from the following grammar where* `[`, `]`, `{`, `}`, `:` *and* `on` *are terminal symbols,* $t \in O_T \cup I_T \cup U_T$*,* $f \in F$*,* $\ell \in$ *Fields, and* α *is a partial mapping from A to Vals:*

$$\phi ::= f\,[\alpha] \mid \ell : f\,[\alpha] \mid \texttt{on}\,t\{\phi\} \mid f\,[\alpha]\{\phi\} \mid \ell : f\,[\alpha]\{\phi\} \mid \phi \dots \phi\ .$$

Example 2.2 *Examples of GraphQL queries conforming to the GraphQL schema* $\mathcal{S}$ *of Example 2.1 are the following:*

$$\phi_1 = \texttt{people(lname: "Doe")}\ \{\ \texttt{fname email}\ \}\ \text{ and}$$
$$\phi_2 = \texttt{companies}\ \{\ \texttt{name employees}\ \{\ \texttt{id lname}\ \}\ \}\ .$$

Both queries demonstrate the hierarchical structure of GraphQL queries. For example, ϕ_2 *accesses fields in the first level that belong to the object type* `Company`*. In the second level, it accesses fields of the object type* `Person`*, as* $\text{type}_{\mathcal{S}}(\texttt{employees}) = \texttt{[Person]}$*.*

GraphQL queries of particular interest are those that are *non-redundant* and in *ground-typed* normal form. According to [22, Theorem 3.8], every GraphQL query can be transformed into an equivalent query that is non-redundant and in ground-typed normal form. An important characteristic of such queries is that their response can be constructed without being subjected to *field collection* [25, Section 6.3.2]. This allows non-redundant GraphQL queries in ground-typed normal form to be evaluated in time linear to the size of their response [22, Corollary 4.3].

Definition 2.4 (GraphQL semantics [22]) *Let* $G = (N,E,\tau,\lambda,\rho)$ *be a GraphQL graph and* ϕ *a non-redundant GraphQL query in ground-typed normal form, both conforming to a GraphQL schema* $\mathcal{S}$ *over* (F,A,T)*. The evaluation of* ϕ *over G from node* $u \in N$*, denoted by* $[\![\phi]\!]^u_G$*, is captured by Equation 1.*[2] *The evaluation of* ϕ *over G, denoted by* $[\![\phi]\!]_G$*, is simply* $[\![\phi]\!]^\rho_G$*.*

[2]The expressions $\ell : f\,[\alpha]\{\phi\}$ and $\ell : f\,[\alpha]$ are evaluated by replacing f with ℓ in the first two rules' results.

$$[\![f\,\texttt{[}\alpha\texttt{]}]\!]_G^u = \begin{cases} f : \lambda(u, f[a]) & \text{if } (u, f[a]) \in dom(\lambda), \\ f : null & \text{else.} \end{cases}$$

$$[\![f\,\texttt{[}\alpha\texttt{]}\,\{\phi\}]\!]_G^u = \begin{cases} f : [\{[\![\phi]\!]_G^{v_1}\} \ldots [\![\phi]\!]_G^{v_k}] & \text{if } type_{\mathcal{S}}(f) \in L_T \text{ and} \\ & \{v_1 \ldots v_k\} = \{v_i \mid (u, f[a], v_i) \in E\}, \\ f : \{[\![\phi]\!]_G^{v}\} & \text{if } type_{\mathcal{S}}(f) \notin L_T \text{ and} \\ & (u, f[a], v) \in E, \\ f : null & \text{if } type_{\mathcal{S}}(f) \notin L_T \text{ and there is no} \\ & v \in N \text{ s.t. } (u, f[a], v) \in E. \end{cases} \tag{1}$$

$$[\![\texttt{on}\; t\{\phi\}]\!]_G^u = \begin{cases} [\![\phi]\!]_G^u & \text{if } t \in O_T \text{ and } \tau(u) = t, \\ \varepsilon & \text{else } (\varepsilon \text{ denotes the empty word).} \end{cases}$$

$$[\![\phi_1 \ldots \phi_k]\!]_G^u = [\![\phi_1]\!]_G^u \ldots [\![\phi_k]\!]_G^u.$$

In this work, we assume that $A \subset F$. More specifically, we restrict the set of arguments of a field $f \in F$ to be the set of scalar fields of its type, i.e., $args_{\mathcal{S}}(f) \subseteq \{f' \mid f' \in fields_{\mathcal{S}}(type_{\mathcal{S}}(f)), type_{\mathcal{S}}(f') \in Scalars\}$. Hence, leaf fields are not assigned any arguments, and the expressions $f\,\texttt{[}\alpha\texttt{]}$ and $\ell : f\,\texttt{[}\alpha\texttt{]}$ can be written as f and $\ell : f$, respectively [22]. In [22], the sets F and A are assumed to be disjoint; however, our assumption is in accordance with the GraphQL specification and does not affect the provided semantics.

2.2. *Worst-case Optimal Multi-way Join Algorithms*

Worst-case optimal multi-way algorithms [26] have recently gained a lot of attention (e.g., [3,27,28,29]) and have demonstrated high performance in evaluating graph pattern queries [27,29,30]. Such algorithms satisfy the AGM bound [31] and their runtime matches the worst-case size of the result of the input query [23,27]. Pair-wise join algorithm carry out join operations on two join operands at a time. Instead, worst-case optimal multi-way algorithms (e.g., Leapfrog Triejoin [32]) are recursive and evaluate input queries on a per variable basis. This evaluation method does not store any intermediate results and allows for solution mappings to be directly written to the result.

3. Evaluation of GraphQL Queries over RDF Graphs

In this section, we introduce the multi-way left-join algorithm that we developed for the native execution of GraphQL queries over RDF graphs. Motivated by recent results on the evaluation of basic graph pattern queries presented in [3,27], the proposed left-join algorithm is inspired by worst-case optimal multi-way join algorithms and evaluates queries on a per variable basis. However, unlike join operations, the reordering of left-join operations is not allowed. Hence, we have to pay attention to the order in which the variables of a query are evaluated. In addition, left-join operations might produce partial solutions (i.e., solutions with `null` values in the context of GraphQL). To respect the order of operations during the evaluation of a query and to ensure that partial solutions will not be discarded, we additionally introduce the *operand dependency graph*. Before introducing the operand dependency graph and the proposed multi-way left-join algorithm, we define first the process of generating the query operands of GraphQL queries.

3.1. GraphQL Query Operands

In the case of SPARQL, there are multiple features of the language that generate query operands, with the most common being the triple pattern. In the case of GraphQL, an operand needs to be generated for each *field*, *argument* and *inline fragment* of a query. Here, for simplicity, we use a notation that resembles SPARQL's triple patterns and present how to generate the operands of GraphQL queries. Note that we do not actually translate GraphQL queries to SPARQL queries. Potential implementations are free to use any means available (e.g., indices) for generating these operands. For the generation of GraphQL query operands, we must also map the types and fields of the provided GraphQL schema to RDF terms. Our implementation computes this mapping using a GraphQL *directive* [25, Section 3.13]. In the following, we omit this mapping for brevity.

In a GraphQL query, we distinguish three types of fields: i) *root* fields, ii) *inner* fields, and iii) *leaf* fields. The root field of a query is the starting point of the traversal. Its corresponding operand should only contain the entities of the underlying RDF graph that are instances of its type. The pattern $\langle ?var, \texttt{rdf:type}, type_{\mathcal{S}}(f)\rangle$ is used to extract these instances, with $?var$ being a variable that will be assigned the extracted instances. Inner and leaf fields represent edges between a source and a target vertex in the graph and their operands are created using patterns of the form $\langle ?var_1, f, ?var_2\rangle$. Ultimately, $?var_1$ will be assigned the source vertices of the edge, whereas $?var_2$ will be assigned the target vertices. In the case of inner and leaf fields, we need to also consider the type of the target vertices. More specifically, in RDF, the objects of properties can vary in type, whereas, in GraphQL, the target vertices of fields are of specific type. To restrict the type of target vertices, an additional operand is generated using the pattern of root fields presented above. In practice, this additional operand can be omitted, if the schema allows it (e.g., via a directive). Provided an expression $f\,\texttt{[}\alpha\texttt{]}\,\texttt{\{}\phi\texttt{\}}$, the operand of an argument-value pair $\mathsf{a} = (f', v) \in \alpha$ is created by $\langle ?var, f', v\rangle$. Last, the operand of an inline fragment $\texttt{on}\; t\texttt{\{}\phi\texttt{\}}$, whose sub-expression ϕ is executed only if the parent field is an instance of the type t, is created by $\langle ?var, \texttt{rdf:type}, t\rangle$. The operands of the aliased fields $\ell : f\,\texttt{[}\alpha\texttt{]}\,\texttt{\{}\phi\texttt{\}}$ and $\ell : f\,\texttt{[}\alpha\texttt{]}$ are generated using the patterns of $f\,\texttt{[}\alpha\texttt{]}\,\texttt{\{}\phi\texttt{\}}$ and $f\,\texttt{[}\alpha\texttt{]}$, respectively.

Two query operands participate in a (left-)join operation, if they share a variable. For assigning variables to operands, we take advantage of the hierarchical structure of GraphQL. More specifically, the target vertices of a field and the source vertices of its nested fields, share the same variable. The operands of inline fragments are also assigned the variable of the target vertices of their parent fields. In the case of arguments, their operands are assigned the variable that is already assigned to the operand of their field. Example 3.1 demonstrates the operand generation process of GraphQL queries.

Example 3.1 *Consider the queries of Example 2.2. The operands of ϕ_1 are generated by the patterns: 1) $\langle ?x, \texttt{rdf:type}, \texttt{Person}\rangle$, 2) $\langle ?x, \texttt{lname}, \texttt{"Doe"}\rangle$, 3) $\langle ?x, \texttt{fname}, ?y\rangle$, and 4) $\langle ?x, \texttt{email}, ?z\rangle$. Note that the operands of the root field and its argument share the same variable. Consequently, vertices representing people whose last name is not "Doe" will be discarded. The inner fields are associated with the root field through the* *variable $?x$. Also note that the target vertices of the inner fields are assigned different variables. The operands of ϕ_2 are created in a similar manner and their corresponding patterns are: 1) $\langle ?x, \texttt{rdf:type}, \texttt{Company}\rangle$, 2) $\langle ?x, \texttt{name}, ?y\rangle$, 3) $\langle ?x, \texttt{employees}, ?z\rangle$, 4) $\langle ?z, \texttt{rdf:type}, \texttt{Person}\rangle$, 5) $\langle ?z, \texttt{lname}, ?w\rangle$, and 6) $\langle ?z, \texttt{id}, ?v\rangle$. The inner fields `name` and `employees` are associated with the root field `companies` through the variable $?x$, whereas the operands of the leaf fields `id` and `lname` are associated with the operand of their parent field, namely `employees`, through $?z$. Last, note the additional operand that is generated for the field `employees`. Its goal is to discard vertices that are not of type `Person`. We assume that type filtering is not required for scalar types for brevity.*

3.2. Operand Dependency Graph

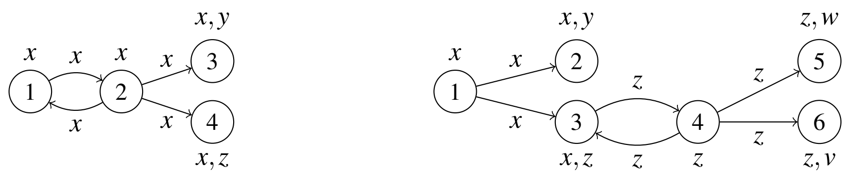
The operand dependency graph is inspired by *pattern trees* [33] and captures the dependencies between the operands of a query. If an operand is not successfully resolved during the query evaluation, its dependent operands should not be evaluated. For example, provided a GraphQL query $f\,\texttt{[}\alpha\texttt{]}\,\{\phi\}$, the operands of ϕ should not be considered if the operands of $f\,\texttt{[}\alpha\texttt{]}$ do not produce any results. However, if the operands of ϕ do not produce any results, the results generated by $f\,\texttt{[}\alpha\texttt{]}$ should not be discarded. The operand dependency graph is formally defined as follows.

Definition 3.1 (Operand dependency graph) *Let O be a list of query operands and Σ an alphabet. Furthermore, let $\mathbb{I}_n = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$. An operand dependency graph is a directed vertex-edge-labelled graph $G = (V, E)$, where $V = \mathbb{I}_{|O|}$ and $E \subseteq V \times \Sigma \times V$. An operand $v \in V$ depends on operand $u \in V$, if and only if $\exists e \in E$ such that $e = (u, \sigma, v)$ and $\sigma \in \Sigma$.*

As per Definition 3.1, the vertices of an operand dependency graph correspond to the operands of its respective query. The variables appearing in query operands are assigned unique labels stemming from Σ and are used to label the vertices and edges of the dependency graph. The vertices of the dependency graph are assigned the labels of their respective operands' variables. Two operands are connected via an edge only if they share a variable. The label of an edge is determined by the label shared by its incident vertices.

For the construction of the operand dependency graph, we take advantage of the hierarchical structure of GraphQL queries. Provided an expression $f\,\texttt{[}\alpha\texttt{]}\,\{\phi\}$ $(\ell : f\,\texttt{[}\alpha\texttt{]}\,\{\phi\})$, the operands of $f\,\texttt{[}\alpha\texttt{]}$ comprise a strong component in the dependency graph, as they all depend on each other. This means that any vertex v of $f\,\texttt{[}\alpha\texttt{]}$ is reachable from any other vertex u of $f\,\texttt{[}\alpha\texttt{]}$, with $v \neq u$, provided that $f\,\texttt{[}\alpha\texttt{]}$ generates multiple operands. As the operands of ϕ depend on the operands of $f\,\texttt{[}\alpha\texttt{]}$, the vertices of $f\,\texttt{[}\alpha\texttt{]}$ are not reachable from the vertices of ϕ. In the case of `on` $t\{\phi\}$ expressions, the operands of ϕ depend on the operand of t. This means that the vertex of t and the vertices of ϕ are connected with edges, whose source is the vertex corresponding to t. In the case of $\phi_1 \ldots \phi_k$ expressions, there are not any edges between the vertices of any ϕ_i and ϕ_j, with $1 \leq i, j \leq k$ and $i \neq j$, as the evaluation of ϕ_i does not affect the evaluation of ϕ_j (Definition 2.4). Operands that depend on each other participate in join operations, whereas unidirectional edges denote left-join operations. Last, in the multi-way left-join algorithm, which is presented below, we make use of the root node of the directed acyclic graph connecting the strongly connected components of an operand dependency graph. Herein, we refer to this node as the *independent strong component* of the dependency graph.

Example 3.2 *The operand dependency graphs corresponding to the GraphQL queries Example 2.2 and their respective operands (Example 3.1) are as follows.*



In the operand dependency graph of ϕ_1 (left-hand side), there are not any edges connecting the vertices of operands 3 *and* 4*, since the evaluation of* 3 *does not affect the evaluation of* 4*, and vice versa. The independent strong component of the dependency graph consists of the operands* 1 *and* 2*. In the operand dependency graph of ϕ_2 (right-hand side), operands* 3 *and* 4 *depend on each other. Both operands are generated by the inner field* `employees`*, with operand* 4 *being responsible for removing any RDF terms assigned to ?z that are not of type* `Person`*. In this case, the independent strong component consists of a single vertex, namely the vertex corresponding to operand* 1.

3.3. Multi-way Left-Join Algorithm

Here, we present our multi-way left-join algorithm (Algorithm 1) for the evaluation of GraphQL queries over RDF graphs. The key characteristics of our approach are the following. First, it evaluates join and left-join operations on a variable simultaneously. Second, it uses the operand dependency graph to eliminate the transitively dependent operands of an empty operand (i.e., an operand that is not successfully resolved), thus avoiding unnecessary operations.

The function `MWLJ` (lines. 1–5) takes as input a GraphQL query and is responsible for generating the operands of the query (line 2) and their dependency graph (line 3). For simplicity, we assume throughout the algorithm that the operands are stored within the vertices of the graph. This function is also responsible for initializing the solution mapping, which stores the bindings of all variables of the query, as its domain is equal to the set of labels appearing in the query's operand dependency graph. Recall that each variable is assigned a unique label (Section 3.2). After initializing the solution mapping, `MWLJ` calls the recursive function `MWLJ_REC` (line 5), which takes the operands dependency graph G and the solution mapping X as inputs.

The function `MWLJ_REC` (lines 6–22) is responsible for carrying out the join and left-join operations and generating the solutions of the query. In case the provided dependency graph is disconnected, `MWLJ_REC` is called for each connected component of the graph (lines 7–9). Disconnected dependency graphs correspond to $\phi_1 \ldots \phi_k$ expressions, as there are no dependencies between any ϕ_i and ϕ_j, with $1 \leq i, j \leq k$ and $i \neq j$ (Section 3.2). If the provided graph is not *strongly* connected, there are left-join operations that need to be carried out (lines 10–20). To respect the order of left-join operations, the algorithm focuses on the set of labels (i.e., variables) U that are found in the independent strong component of the dependency graph (line 11). For the GraphQL queries that we consider in this work, the set U contains only a single label. This will not be the case once we take GraphQL's *input object types* [25, Section 3.10] into consideration.[3] After

[3]Note that also the formal definitions of GraphQL in [22] do not consider input object types.

Algorithm 1 Multi-way Left-join Algorithm

1: **function** MWLJ(Q) ▷ Q: Input GraphQL query
2: $O \leftarrow$ generate the operands of Q
3: $G \leftarrow$ create the operand dependency graph of Q using O
4: $X \leftarrow$ initialize solution mapping with domain equal to the set of labels appearing in G
5: MWLJ_REC(G, X)
6: **function** MWLJ_REC(G, X) ▷ G: operand dependency graph, X: solution mapping
7: **if** G is disconnected **then** ▷ Evaluation of $\phi_1 \dots \phi_k$ expressions
8: **for all** connected components G_i of G **do** ▷ Each G_i corresponds to a ϕ_i, $1 \leq i \leq k$
9: MWLJ_REC(G_i, X)
10: **else if** G is not *strongly connected* **then** ▷ Left-join operation
11: $U \leftarrow$ the set of labels appearing in the independent strong component of G
12: $x \leftarrow$ select a label from U
13: **for all** values χ of x **do**
14: resolve x in all operands using χ ▷ Carries out join and left joins simultaneously
15: $G' \leftarrow$ prune vertices of empty operands and their transitively dependent vertices from G
16: **if** G' is empty **then**
17: **continue** ▷ All operands are pruned (unsuccessful join); continue with the next χ
18: update the value of x in X with χ ▷ Join operations were successful
19: remove x from G'; remove vertices without any labels from G'
20: MWLJ_REC(G', X)
21: **else** ▷ G is strongly connected (no left-join operations)
22: MWJ(G, X) ▷ Carry out multi-way join (no more left joins after this point)

selecting a label x from U, the algorithm iterates over all possible values of x and carries out all join and left-join operations on x (line 14). The algorithm proceeds by removing any operands that were not successfully resolved along with their transitively dependent operands, which can be found by traversing the dependency graph (line 15). If the resulting graph G' ends up being empty, a join operation was not successful and the algorithm continues with the next value of x (lines 16–17). If G' is not empty, the solution mapping X is updated with the current value of x, which is removed from G' along with any fully resolved operands, and the algorithm proceeds with the next recursive step (lines 18–20). In case the provided graph G is *strongly* connected, the algorithm proceeds with a multi-way join algorithm, as there are no left-join operations left to be carried out. The active solution mapping X will be ultimately projected once the remaining join operations are carried out by the multi-way join algorithm.

Example 3.3 *Consider the query ϕ_1 of Example 2.2. Provided the example RDF graph*

```
<p1> rdf:type <Person>; <lname> "Doe"; <fname> "Jon"; <email> "e1".
<p2> rdf:type <Person>; <lname> "Doe"; <fname> "Jan".
```

the proposed algorithm will produce three solutions: {`x:p1, y:"Jon"`}, {`x:p1, z:"e1"`}, *and* {`x:p2, y:"Jan"`}. *The algorithm selects first the label corresponding to the variable* `x`*, which is assigned the identifiers of people in the graph. For the value* `p1` *of* `x`*, the algorithm generates two solutions. The first one provides the first name (*`fname`*) of* `p1`*, which is assigned to* `y`*, whereas the second one provides its email, which is assigned to* `z`*. For the value* `p2`*, the algorithm generates only one solution, as* `p2` *does not have an email in the example graph. Note that after selecting x and removing it from the operand dependency graph, the resulting dependency graph is disconnected. Variables that do not appear in a solution mapping are unbound in that particular mapping.*

Regarding the enumeration of GraphQL queries, in [22], the authors study the enumeration problem for GraphQL queries that are non-redundant and in ground-typed normal form. Recall that such queries can be computed in time linear to the size of their response (Section 2.1). Each solution mapping generated by our algorithm captures a unique path of the response corresponding to the provided query. As our left-join algorithm computes a solution mapping entirely, we are able to directly construct the path that corresponds to a particular solution mapping, once it is evaluated. In addition, due to the recursive nature of our algorithm, the solution mappings of the sub-trees of a particular node of a GraphQL response share common values (Example 3.3). Hence, we are able to avoid visiting the nodes of a response multiple times.

3.4. Implementation

We have implemented the proposed algorithm within the tensor-based triple store Tentris [3]. Tentris achieves state-of-the-art performance in the evaluation of basic graph patterns, which are evaluated by a worst-case optimal multi-way join algorithm [3,34]. Our implementation, namely TentrisGQL, uses Tentris' multi-way join algorithm (Algorithm 1, line 22), and tensor slicing operations to generate the operands of GraphQL queries.

To bridge the gap between GraphQL schemata and RDF graphs, we follow Neo4j's example[4] and define several *directives* in our implementation. As per the GraphQL specification, "directives can be used to describe additional information for types, fields, fragments and operations" [25, Section 3.13]. We mentioned in Section 3.1 that GraphQL types and fields need to be mapped to RDF terms. To this end, we define in our implementation the directive `@uri`. For example, the type definition `type Person @uri(value: "http://www.exmpl.org/Person")` maps the type `Person` to the RDF term `http://www.exmpl.org/Person`. As the inverse of a property is not always available in RDF graphs, we also define the field directive `@inverse`, which denotes that the inverse direction of a field's property should be used. Last, we introduce the field directive `@filter`, which denotes that the results of a particular field should be filtered using that field's type. This directive should be used on fields that are mapped to properties having ranges consisting of multiple RDF classes (Section 3.1).

4. Experimental Results

In this section, we present the performance evaluation of TentrisGQL, which we evaluated using the Linköping GraphQL Benchmark (LinGBM) [24]. LinGBM is a synthetic benchmark generator that provides a GraphQL schema that captures the structure of the generated datasets, and a set of 16 GraphQL query templates. To the best of our knowledge, LinGBM is currently the only publicly available benchmark for evaluating GraphQL services. The experiments that are presented below were carried out on a Debian 10 server with an AMD EPYC 7742 64-Core Processor, 1TB RAM, and two 3 TB NVMe SSDs in RAID 0. All artifacts (e.g., datasets, GraphQL schemata, queries, and system configurations) are available online.[5]

[4] https://github.com/neo4j-graphql/neo4j-graphql-js/blob/master/docs/graphql-schema-directives.md

[5] https://github.com/dice-group/graphql-benchmark

4.1. Systems

As baseline for our experiments, we used Neo4j Community Edition 5.5.0 [11]. We selected Neo4j because it is a widely used graph database and it provides its own tools for processing GraphQL queries. In our experiments, we evaluated Neo4j in two different modes. In the first mode (Neo4jC), Neo4j was provided with Cypher queries instead of GraphQL queries. The GraphQL queries used in our experiments (Section 4.2) were translated to Cypher queries using a library provided by Neo4j[6]. The purpose of this mode was to compare the query evaluation performance of TentrisGQL against that of Neo4j, as no result rewriting takes place in this mode of Neo4j. To find out the overhead introduced by the process of result rewriting, we used a second mode, namely Neo4jGQL. Neo4jGQL includes an external application that is connected to Neo4j and is responsible for translating GraphQL queries to Cypher queries and rewriting query results to GraphQL responses.[7] Recall that TentrisGQL incrementally constructs GraphQL responses. For the evaluation, we used Neo4j's recommended memory settings[8] and built the appropriate search indices. More specifically, regarding the memory settings, we allocated 31GB of memory to the Java virtual machine (JVM) and 957GB for caching purposes. In our experiments, we also evaluated TentrisGQLBase, a version of TentrisGQL that treats fields of type ID (i.e., fields that capture IRIs of RDF terms) as strings. As a result, TentrisGQLBase needs to carry out left joins and joins to evaluate such fields when they appear as leaf fields or arguments in a query, respectively. In contrast, TentrisGQL accesses the IRIs of RDF terms directly. TentrisGQLBase provides us with insights on the impact that the evaluation of leaf fields has on the performance of our service.

4.2. Datasets, Query Templates, and Schema

LinGBM's dataset generator relies on the dataset generator of LUBM [35] and allows for the generation of datasets of varying sizes via the use of a scale factor. To evaluate the performance of our approach on RDF graphs of different sizes, we generated three graphs (Table 1), namely *LinGBM100*, *LinGBM500*, and *LinGBM1000*. For our experiments, we modified LinGBM's dataset generator to include the classes corresponding to the interface types of the schema, as both systems expect them to be stated in the input data.

As previously mentioned, LinGBM also provides a set of query templates. Their design follows a choke-point methodology [24, Section 3.3]; each choke-point focuses on a particular workload or operation. In this work, we are interested in join and left-join operations. Hence, we focus on the choke-points *Attribute Retrieval* (CP1) and *Relationship Traversal* (CP2) of LinGBM. There are only six query templates (QT1-QT6) that focus exclusively on CP1 and CP2. To include additional queries in our evaluation, we modified the query templates QT7-QT14 by removing those features that are not related to CP1 and CP2 (e.g., ordering, filtering, and pagination). In addition, we had to remove input objects from the query templates QT11-QT14, as they are currently not supported by our implementation. Ultimately, in our experiments, we used 11 query templates and two non-parameterized queries (Table 2).[9]

[6] `https://github.com/neo4j-graphql/neo4j-graphql-js`
[7] We followed the example used in `https://github.com/neo4j-graphql/neo4j-graphql-js`.
[8] `https://neo4j.com/docs/operations-manual/5/tools/neo4j-admin/neo4j-admin-memrec`
[9] After our modifications, QT8 and QT11 do not have any parameters, and QT13 and QT14 are identical.

	Scale Factor	#Triples	#Distinct Subjects	#Distinct Predicates	#Distinct Objects
LinGBM100	100	16M	2M	20	3M
LinGBM500	500	79M	10M	20	18M
LinGBM1000	1000	160M	21M	20	37M

Table 1. The datasets used in the experiments.

QT	D	aRS-100	aRS-500	aRS-1000	QT	D	aRS-100	aRS-500	aRS-1000
1	3	34K	170K	338K	8	1	4M	20M	40M
2	3	21K	105K	200K	9	4	279K	1.3M	2.7M
3	4	243	244	245	10	1	6.3M	31M	63M
4	5	81K	425K	864K	11	2	3.7M	18M	37M
5	7	12M	311M	1.2G	12	3	79K	397K	785K
6	4	19K	94K	192K	13	3	73K	373K	738K
7	3	20K	101K	202K					

Table 2. The depth (D) and the average size of the GraphQL response (aRS-SF) in bytes of each GraphQL query template (QT) for each scale factor (SF). QT8 and QT11 are not parameterized.

The GraphQL schema provided by LinGBM is meant to be used by GraphQL services that do not generate GraphQL schemata automatically (e.g., relational schema to GraphQL schema). For our experiments, we modified the provided schema by removing those types and features that are not required by the query templates (e.g., input types and enumeration types) used in the experiments. For TentrisGQL, we extended the schema with the directives of our implementation (Section 3.4). In a similar manner, we extended the schema used for Neo4j with Neo4j's respective directives.

4.3. Benchmark Configurations and Execution

Our GraphQL service was evaluated on two different benchmark configurations. The purpose of the first configuration was to evaluate the performance of our service on each query template. For each template, we created a stress test consisting of ten query instances per template (110 queries) We also created a stress test for each non-parameterized query (112 queries in total). The stress tests were executed 5 consecutive times and independently from each other, thus ensuring that the query instances were executed the same number of times. With the second configuration, we measured the performance of our system when queried by multiple clients. This configuration consisted of one stress test, which included one query instance from each template and the two non-parameterized queries (i.e., 13 queries). During the execution of the second configuration, it was important that there were multiple clients issuing queries at all times. For this reason, we configured the clients to issue queries concurrently for one hour [3,36]. In both configurations, the execution of each stress test is preceded by a warm-up run, in which the queries of the corresponding stress test are executed once. This allowed Neo4j to load and cache its data structures in the main memory.

The stress tests of both benchmark configurations were executed over HTTP using the benchmark execution framework IGUANA in version 3.3.0 [36]. As in [3], we set

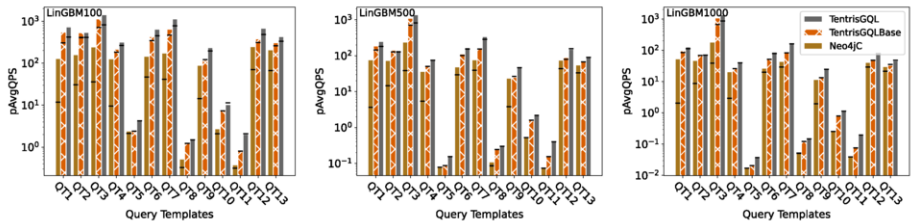


Figure 1. Performance of the systems in the first configuration w.r.t. their pAvgQPS. The black lines denote the values reported in the warmup run.

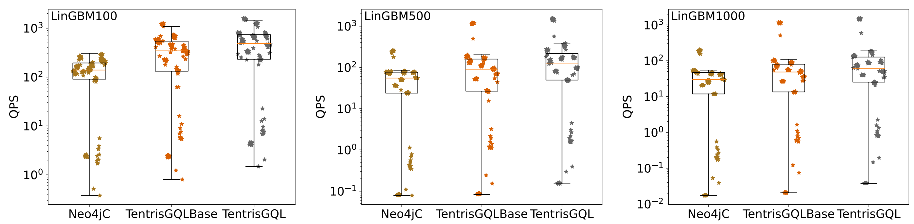


Figure 2. Performance of the systems in the first configuration w.r.t. their QPS.

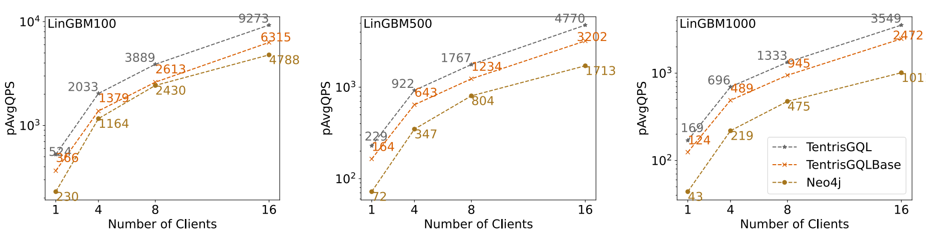


Figure 3. Scalability of the systems in the second configuration w.r.t. their pAvgQPS.

	Neo4jC	Neo4jGQL	TentrisGQLBase	TentrisGQL
LinGBM100	230.32	91.51	366.27	524.90
LinGBM500	72.13	33.72	164.23	229.90
LinGBM1000	43.99	20.11	124.59	169.79

Table 3. Overhead of result rewriting in Neo4jGQL (pAvgQPS).

the timeout across all benchmarks to three minutes and measured the performance of our implementation using the number of queries executed per second (QPS) and the penalized average QPS (pAvgQPS); the penalty for failed queries (e.g., timed out queries) was set to three minutes. Last, we compared the results generated by all systems to ensure that they return the same results across all queries.

4.4. Results

The results of the first benchmark configuration are presented in Figures 1 and 2. Figure 1 shows that both TentrisGQL and TentrisGQLBase outperform Neo4jC across all query templates in all datasets, with TentrisGQL achieving 1.5 (QT2 and QT13) to 7.4 (QT3) times higher pAvgQPS than Neo4jC in the largest dataset, namely LinGBM1000. In addition, both TentrisGQL and TentrisGQLBase achieve higher median QPS than Neo4jC in all datasets (Figure 2). Figure 3 summarizes the results reported in the second benchmark configuration. We removed the query instance corresponding to QT5 from the second configuration's query list because Neo4j was running out of JVM memory when this query was issued by multiple clients. TentrisGQL and TentrisGQLBase did not face any memory-related issues. Figure 3 shows that Neo4jC scales better than TentrisGQL and TentrisGQLBase when queried by 4 and 8 concurrent clients in the smallest dataset (i.e., LinGBM100). However, TentrisGQL and TentrisGQLBase achieve higher pAvgQPS than Neo4jC in all cases and in particular, TentrisGQL achieves 3.5 higher pAvgQPS in the case of 16 clients in LinGBM1000. To measure the overhead introduced by the rewriting of Neo4j's results to GraphQL responses, we used the second benchmark configuration with one concurrent user. Table 3 shows that rewriting process leads to TentrisGQL achieving up to 8.4 times higher pAvgQPS than Neo4jGQL

4.5. Discussion

The performance of the systems did not vary significantly across all datasets. In particular, they were not significantly affected by the increasing size of the datasets. The systems' performance was mostly affected by the average result size (aRS) of the query templates (Table 2). In particular, all systems achieved their highest and lowest pAvgQPS in all datasets in QT3 and QT5, respectively. QT3 has the lowest aRS, whereas QT5 has the highest. The depth of the query templates also affects the systems' performance. For example, the pAvgQPS of the systems in QT6 is lower than in QT7, even though the latter has a higher aRS. Queries with higher values of depth require more left-join operations in TentrisGQL and longer path traversals in Neo4j.

Another factor that impacts the performance of our GraphQL service is the size of the operands corresponding to leaf fields, which are evaluated via left-join operations. This observation is grounded in the performance of TentrisGQLBase (Figure 1), which is always equivalent to, or worse than, TentrisGQL's performance. Recall that TentrisGQLBase, unlike TentrisGQL, evaluates leaf fields and arguments of type ID via left-join and join operations, respectively. Neo4j employs the property graph model, which allows it to represent leaf fields as node properties. Hence, for evaluating leaf fields, Neo4j does not iterate over all of a particular property key's properties. Despite these additional operations, both TentrisGQL and TentrisGQLBase outperform Neo4j. This suggests that our algorithm does not introduce much overhead to the computations.

The results of Table 3 are in line with the results reported in [18] and demonstrate the importance of GraphQL services being able to directly construct GraphQL responses. Regarding the memory usage, we measured the memory used by the systems in LinGBM1000 when queried by 16 concurrent clients using `pmap`[10]. The highest Resident Set Size (RSS) reported by TentrisGQL and Neo4jC was 41GB and 44GB, respectively.

[10] https://linux.die.net/man/1/pmap

5. Related Work

Recently, several graph storage solutions have made efforts to allow users to access their data via GraphQL. Dgraph [16] is a distributed graph database that natively supports GraphQL. It also provides its own query language, namely DQL. In Dgraph, GraphQL operations are translated to DQL operations. However, response objects are constructed following the GraphQL specification. Hence, a rewriting of the results is not required. We did not include Dgraph in our experiments for two reasons. First, Dgraph does not fully support RDF, as it is not able to handle URIs. Additionally, Dgraph's GraphQL service expects predicates to be prefixed with their subject's type. Consequently, existing RDF graphs need to be substantially modified to be stored in a Dgraph instance. Second, Dgraph does not provide a bulk loader for its GraphQL service; hence it is not able to load large knowledge graphs efficiently.[11] In addition to the translation tools used in Section 4, Neo4j provides a library that serves as a middleware between applications and database instances. This library[12] is responsible for the translation process of GraphQL queries to Cypher queries. Regarding triple stores, Stardog [37] and the commercial edition of GraphDB [9] provide GraphQL support by translating GraphQL to SPARQL. Virtuoso [8] introduced a GraphQL plugin[13] that allows its users to query RDF graphs via GraphQL. To bridge the gap between GraphQL and SPARQL, this plugin relies on OWL ontologies to map the types and fields of GraphQL schemata to RDF terms. We did not include Virtuoso in our experiments as it does not perform type filtering in inner fields, which leads to queries returning incorrect results.[14]

6. Conclusion and Future Work

We presented an approach for the native evaluation of GraphQL queries over RDF graphs. As GraphQL queries require left-join operations, we focused on the development of a novel multi-way left-join algorithm that is inspired by worst-case optimal multi-way join algorithms. Similarly to worst-case optimal multi-way join algorithms, the proposed left-join algorithm recursively evaluates queries on a per variable basis, which allows for the incremental enumeration of GraphQL queries. By implementing our approach within the tensor-based triple store Tentris, we provide the first publicly available triple store that treats GraphQL as a first-class citizen. The performance evaluation of our implementation demonstrates the efficiency of the left-join algorithm, as our implementation outperforms a state-of-the-art graph database, namely Neo4j.

Our implementation currently supports the features of the language that are required by its formal semantics (Equation 1). Our future work will focus on extending our GraphQL service with all features from the specification. To the best of our knowledge, there have not been any works that focus on evaluating SPARQL queries requiring left-join operations on a variable basis. To this end, we plan to use our approach for the evaluation of such SPARQL queries (i.e., queries containing optional graph patterns).

[11] https://discuss.dgraph.io/t/graphql-vs-dql-dgraph-blog/14311; see paragraph "When not to use GraphQL".

[12] https://neo4j.com/docs/graphql-manual/current/

[13] https://community.openlinksw.com/t/introducing-native-graphql-support-in-virtuoso/3378

[14] https://github.com/openlink/virtuoso-opensource/issues/1115

References

[1] Weikum G, Dong XL, Razniewski S, Suchanek FM. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. Found Trends Databases. 2021:108-490.

[2] Hogan A, Blomqvist E, Cochez M, d'Amato C, de Melo G, Gutiérrez C, et al. Knowledge Graphs. ACM Comput Surv. 2021:71:1-71:37.

[3] Bigerl A, Conrads F, Behning C, Sherif MA, Saleem M, Ngomo AN. Tentris - A Tensor-Based Triple Store. In: The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I; 2020. p. 56-73.

[4] Zou L, Özsu MT, Chen L, Shen X, Huang R, Zhao D. gStore: a graph-based SPARQL query engine. VLDB J. 2014:565-90.

[5] Yuan P, Liu P, Wu B, Jin H, Zhang W, Liu L. TripleBit: a Fast and Compact System for Large Scale RDF Data. Proc VLDB Endow. 2013:517-28.

[6] Neumann T, Weikum G. RDF-3X: a RISC-style engine for RDF. Proc VLDB Endow. 2008;1(1):647-59.

[7] Systap LLC. Blazegraph; N.D. Retrieved 2023-05-03. https://blazegraph.com/.

[8] Openlink. Virtuoso; N.D. Retrieved 2023-05-03. http://vos.openlinksw.com/owiki/wiki/VOS.

[9] Ontotext. GraphDB; N.D. Retrieved 2023-05-03. https://www.ontotext.com/products/graphdb/.

[10] Apache Software Foundation. Apache Jena Fuseki; N.D. Retrieved 2023-05-03. https://jena.apache.org/documentation/fuseki2/.

[11] Inc N. Neo4j; N.D. Retrieved 2023-03-21. https://neo4j.com/download-center/#community.

[12] The Linux Foundation. JanusGraph; N.D. Retrieved 2023-05-03. https://janusgraph.org/.

[13] Bonifati A, Fletcher GHL, Voigt H, Yakovets N. Querying Graphs. Synthesis Lectures on Data Management. Morgan & Claypool Publishers; 2018.

[14] Francis N, Green A, Guagliardo P, Libkin L, Lindaaker T, Marsault V, et al. Cypher: An Evolving Query Language for Property Graphs. In: Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018; 2018. p. 1433-45.

[15] Rodriguez MA. The Gremlin graph traversal machine and language (invited talk). In: Proceedings of the 15th Symposium on Database Programming Languages, Pittsburgh, PA, USA, October 25-30, 2015. Pittsburgh, PA, USA; 2015. p. 1-10.

[16] io D. Dgraph; N.D. Retrieved 2023-05-02. https://dgraph.io/docs. Available from: https://dgraph.io/docs.

[17] Hartig O, Hidders J. Defining Schemas for Property Graphs by using the GraphQL Schema Definition Language. In: Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), Amsterdam, The Netherlands, 30 June 2019; 2019. p. 6:1-6:11.

[18] Gleim LC, Holzheim T, Koren I, Decker S. Automatic Bootstrapping of GraphQL Endpoints for RDF Triple Stores. In: Joint Proceedings of Workshops AI4LEGAL2020, NLIWOD, PROFILES 2020, QuWeDa 2020 and SEMIFORM2020 Colocated with the 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, November, 2020; 2020. p. 119-34.

[19] Chaves-Fraga D, Priyatna F, Alobaid A, Corcho Ó. Exploiting Declarative Mapping Rules for Generating GraphQL Servers with Morph-GraphQL. Int J Softw Eng Knowl Eng. 2020:785-803.

[20] Taelman R, Sande MV, Verborgh R. GraphQL-LD: Linked Data Querying with GraphQL. In: Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018; 2018. .

[21] Semantic Integration LTD. HyperGraphQL; N.D. Retrieved 2023-05-03. https://github.com/hypergraphql/hypergraphql.

[22] Hartig O, Pérez J. Semantics and Complexity of GraphQL. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018; 2018. p. 1155-64.

[23] Ngo HQ. Worst-Case Optimal Join Algorithms: Techniques, Results, and Open Problems. In: Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018; 2018. p. 111-24.

[24] Cheng S, Hartig O. LinGBM: A Performance Benchmark for Approaches to Build GraphQL Servers. In: Chbeir R, Huang ZH, Silvestri F, Manolopoulos Y, Zhang Y, editors. Web Information Systems Engineering - WISE 2022 - 23rd International Conference, Biarritz, France, November 1-3, 2022, Proceedings. vol. 13724 of Lecture Notes in Computer Science. Springer; 2022. p. 209-24.

[25] GraphQL Specification; N.D. Retrieved 2023-03-13. https://spec.graphql.org/June2018.

[26] Ngo HQ, Porat E, Ré C, Rudra A. Worst-case Optimal Join Algorithms. J ACM. 2018:16:1-16:40.

[27] Hogan A, Riveros C, Rojas C, Soto A. A Worst-Case Optimal Join Algorithm for SPARQL. In: The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I. Springer; 2019. p. 258-75.

[28] Arroyuelo D, Hogan A, Navarro G, Reutter JL, Rojas-Ledesma J, Soto A. Worst-Case Optimal Graph Joins in Almost No Space. In: SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021; 2021. p. 102-14.

[29] Freitag MJ, Bandle M, Schmidt T, Kemper A, Neumann T. Adopting Worst-Case Optimal Joins in Relational Database Systems. Proc VLDB Endow. 2020:1891-904.

[30] Kalinsky O, Hogan A, Mishali O, Etsion Y, Kimelfeld B. Exploration of Knowledge Graphs via Online Aggregation. In: 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022. IEEE; 2022. p. 2695-708.

[31] Atserias A, Grohe M, Marx D. Size Bounds and Query Plans for Relational Joins. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA. IEEE Computer Society; 2008. p. 739-48.

[32] Veldhuizen TL. Triejoin: A Simple, Worst-Case Optimal Join Algorithm. In: Proc. 17th International Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014; 2014. p. 96-106.

[33] Letelier A, Pérez J, Pichler R, Skritek S. Static analysis and optimization of semantic web queries. In: Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012; 2012. p. 89-100.

[34] Bigerl A, Conrads L, Behning C, Saleem M, Ngomo AN. Hashing the Hypertrie: Space- and Time-Efficient Indexing for SPARQL in Tensors. In: Sattler U, Hogan A, Keet CM, Presutti V, Almeida JPA, Takeda H, et al., editors. The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings. vol. 13489 of Lecture Notes in Computer Science. Springer; 2022. p. 57-73.

[35] Guo Y, Pan Z, Heflin J. LUBM: A benchmark for OWL knowledge base systems. J Web Semant. 2005:158-82.

[36] Conrads F, Lehmann J, Saleem M, Morsey M, Ngomo AN. Iguana: A Generic Framework for Benchmarking the Read-Write Performance of Triple Stores. In: The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II. Springer; 2017. p. 48-65.

[37] Stardog Union. Stardog; N.D. Retrieved 2023-05-03. https://docs.stardog.com/.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230008

Evaluating Reification with Multi-Valued Properties in a Knowledge Graph of Licensed Educational Resources

Manoé KIEFFER [a,1], Ginwa FAKIH [a] Patricia SERRANO ALVARADO [a]
[a] *LS2N, UMR 6004, Nantes Université, 44300 Nantes, France*

Abstract. This paper presents the construction of a Knowledge Graph (KG) of Educational Resources (ER), where RDF reification is essential. The ERs are described based on the subjects they cover considering their relevance. RDF reification is used to incorporate this subject's relevance. Multiple reification models with distinct syntax and performance implications for storage and query processing exist. This study aims to experimentally compare four statement-based reification models with four triplestores to determine the most pertinent choice for our KG. We built four versions of the KG. Each version has a distinct reification model, namely standard reification, singleton properties, named graphs, and RDF-star, which were obtained using RML mappings. Each of the four triplestores (Virtuoso, Jena, Oxigraph, and GraphDB) was setup four times (except for Virtuoso, which does not support RDF-star), and seven different SPARQL queries were experimentally evaluated. This study shows that standard reification and named graphs lead to good performance. It also shows that, in the particular context of the used KG, Virtuoso outperforms Jena, GraphDB, and Oxigraph in most queries. The recent specification of RDF-star and SPARQL-star sheds light on statement-level annotations. The empirical study reported in this paper contributes to the efforts towards the efficient usage of RDF reification. In addition, this paper shares the pipeline of the KG construction using standard semantic web technologies.

Keywords. Knowledge graph, RDF reification, multi-valued properties, query evaluation, educational resources.

1. Introduction

When teachers want to create a new course, they typically do a keyword search for (open) Educational Resources (ER) on the web to reuse and integrate into their course. While there are numerous valuable and relevant resources available (such as slides, videos, figures, text, code, etc.), many remain undiscovered because they are not well connected. Moreover, using these resources can present legal challenges if their licenses are not compatible with the course's license. These legal issues can create barriers for both the teacher and the institution hosting the course. Ideally, the process of analysing available resources to match a course plan and verifying licenses should not be time-consuming. In our project, the goal is to design a solution that can identify a minimal, relevant set of educational resources with licenses that can protect such set of resources, whether

[1]Corresponding Author: Manoé Kieffer, e-mail: Manoe.Kieffer@univ-nantes.fr.

or not the licenses are open. Our aim is to help teachers create content reusing relevant resources and without having to focus on licensing aspects.

ERs can be described by their title, authors, language, license, etc., as well as the subjects they cover. ERs' subjects can be numerous but not equally relevant for the ER. Some subjects are the main focus, while others are only mentioned briefly. Therefore, the relevance of each subject should be identified, and their relationship with each ER should be weighed accordingly. The best way to make ERs findable and reusable is to use the principles of the Linked Data. Semantic web technologies will allow a detailed description and interconnection of ERs/The recent specification of RDF-star and SPARQL-star sheds light on statement-level annotations. One of the first public work-drafts of RDF 1.2, introduces quoted triples as another kind of RDF term which can be used as the subject or object of another triple[2]. In our particular use case, statement-level reification will allow annotating with scores the relation of ERs and the subjects they treat. As the number of subjects can be important, this reified relation is a multi-valued property. Thus, efficiently dealing with multi-valued properties is important as well.

Multiple reification models with distinct syntax and performance implications for storage and query processing exist. The main objective of this work is to experimentally compare four statement-based reification models on four triplestores to determine the most pertinent choice for our KG.

The contributions of this paper are twofold: (i) a methodology to build four versions of a knowledge graph of ERs using statement-level reification, namely standard reification, singleton property, named graphs and RDF-star, and (ii) an empirical evaluation of four triplestores (Virtuoso, Jena, GraphDB, and Oxigraph) with a set of seven SPARQL query templates grounded with up to six different instances (26 instantiated queries).

The rest of this paper is organised as follows. Section 2 explains the methodology we follow to build the KG of ERs. Section 3 explains the pipeline used. Section 4 evaluates experimentally the reification models. Section 5 describes the related works. Finally, Section 6 outlines our future work and concludes.

2. Knowledge graph description

Our project, aims to empower teachers to facilitate the creation of licensable ERs based on existing ones. The resources in our KG comprise unstructured ERs (documents, videos, and audio files, etc.), which are semantically annotated with DBpedia resources. By means of a wikification process, relevant DBpedia concepts related to ERs are used to provide a comprehensive description of each resource. This section introduces the used ontology (Section 2.1), an explanation of the wikification process (Section 2.2), and statistics on our KG (Section 2.3).

2.1. Used ontology

Figure 1 depicts the used ontology. Consistent with the IEEE LOM standard (Learning Object Metadata)[3], we define ERs as LOM Learning Objects. The LOM standard suggests a range of properties to describe learning objects, using common vocabularies such as Dublin Core and FOAF (dct:title, dct:creator, dct:language, dct:licence, dct:format, foaf:name, etc.).

[2] https://w3c.github.io/rdf-concepts/spec/#section-triples

[3] http://data.opendiscoveryspace.eu/ODS_LOM2LD/ODS_SecondDraft.html

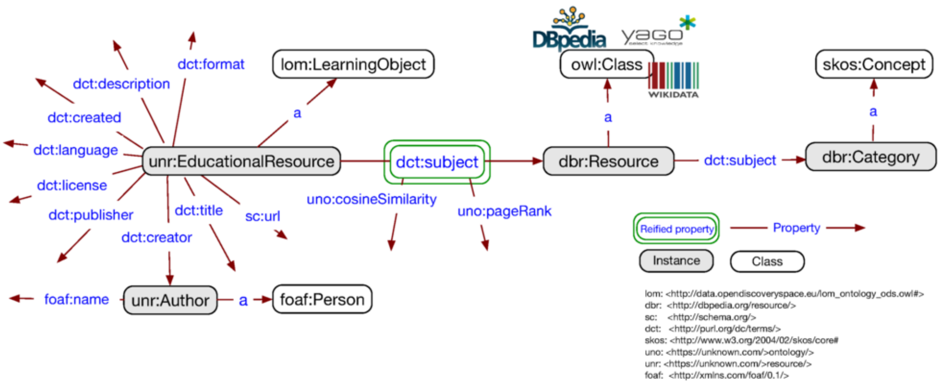


Figure 1. KG ontology.

The particularity of our ontology lies in the extension of the LOM description to consider the subjects treated in the learning objects with relevance scores. To do this, we use RDF statement-level reification. Reification allows making statements about statements in a generic manner. For us, it will allow to state that an ER treats a particular subject (in our case a DBpedia resource) *to some extent*. Concretely, it will allow to annotate the property dct:subject in a fact (unr:EducationalResource, dct:subject, dbr:Resource) with relevance scores. These scores are determined with a wikification process, which identifies pageRank and cosine similarity values in the range $[0..1]$. More information on this process is provided in the next section.

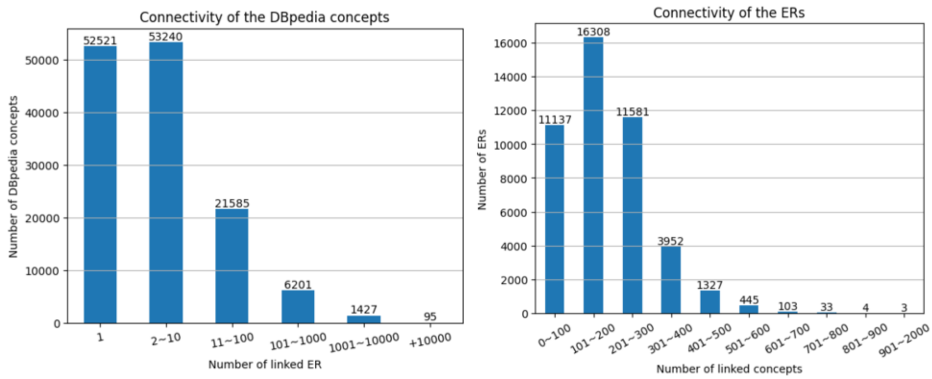
Besides being reified, dct:subject is a multi-valued property, i.e., a subject-predicate pair having several objects. These objects are DBpedia resources which are instances of classes in DBpedia, Wikidata, and Yago ontologies. We consider also DBpedia categories, which are used in Wikipedia to organize articles and pages by subject matter. Since the goal of our KG is to identify ERs based on their subjects, DBpedia categories are essential. DBpedia resources are associated with their categories through the dct:subject property (dbr:Resource, dct:subject, dbr:Category).

2.2. *Wikification of educational resources*

Entity linking techniques that map named entities to Wikipedia entities are called wikification. Text wikification is the task of automatically extracting the most important words and phrases in a document, and identifying for each such keyword the appropriate link to a Wikipedia article [1]. The wikification process generally involves two phases: term extraction and link disambiguation.

There are various approaches to wikification that differ in the techniques used for extracting phrases and linking them to external resources. The wikification tool called *Wikifier* has shown a good performance compared to some state-of-the-art approaches [2][4]. This tool identifies *mentions* - phrases extracted from the input document - and uses them as hyperlinks between Wikipedia pages. The Wikipedia pages linked by a mention are considered as candidate concepts for that mention. Wikifier constructs a bipartite graph consisting of the mentions and their corresponding candidate concepts. The internal structure of hyperlinks between Wikipedia pages is then leveraged to weigh the edges

[4] https://wikifier.org



(a) Connectivity of the DBpedia concepts.

(b) Connectivity of ERs.

Figure 2. Statistics of the connectivity in the graph.

(mentions) of the bipartite graph. A mention may have several candidate concepts because the same text can lead to different Wikipedia pages. To disambiguate mentions, the pageRank algorithm is applied over the graph. The concept with the highest pageRank score is selected for each mention, resulting in a set of Wikipedia concepts representing the input document. A threshold is then applied to retrieve the top-ranking concepts. In addition, Wikifier calculates the cosine similarity between the input document and the Wikipedia pages of the top-ranking concepts.

Currently, our project has collected a set of open educational resources that were wikified in the X5GON project[5].

2.3. Statistics and dataset content

There exist several vocabularies to describe RDF datasets. The VoID vocabulary [3] is the most well-established vocabulary.Additionally, the DCAT vocabulary [4] is a W3C recommendation to describe datasets, data services and data catalogs.We provide a description of our KG using VoID and DCAT in a VoID file[6]. The metadata for the dataset includes the label, license, SPARQL endpoint, provenance, prefixes used, and general statistics such as the number of triples, entities, subjects, objects, properties, etc. Our class partition currently consists of roughly 45K learning objects (lom:LearningObject), 13K authors (foaf:Person), twelve licenses (odrl:Policy), and 2,2M categories (skos:Concepts)[7] along with 135K DBpedia resources that serve as the reified concepts of ERs. It is important to note that reification is employed to annotate the dct:subject relation with two different annotations. The annotations are the pageRank score and cosine similarity score discussed in Section 2.2, those are numerical values between 0 and 1.

The distribution of the reified subjects (called concepts from now on) over ERs is far from being uniform. We consider the *connectivity of a concept* as the number of multi-valued properties (dct:subject) linking to it. Figure 2a shows that 100K over 135K

[5] https://www.x5gon.org

[6] https://gitlab.univ-nantes.fr/-/ide/project/clara/pipeline/tree/main/-/VoIDstatistics/clara-metadata.ttl/

[7] We obtained the entire hierarchy of DBpedia categories.

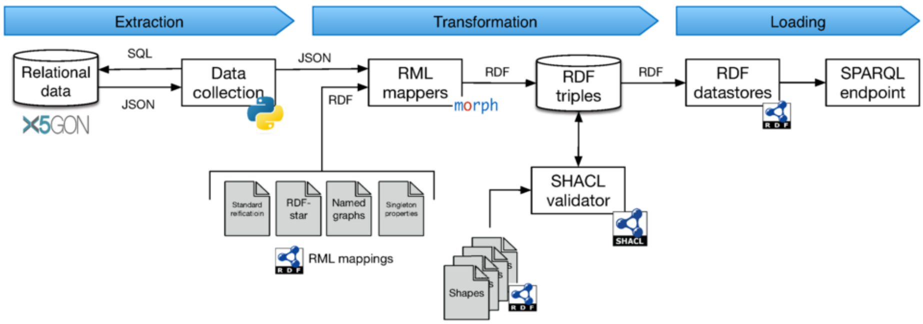


Figure 3. Pipeline of the ETL (Extract, Transform, Load) process.

concepts have poor connectivity. Roughly 52K concepts are associated with a single ER, while around 53K are connected to between 2 and 10 ERs, as shown in the first two columns. The third column indicates that around 21K concepts connect between 11 and 1000 ERs. Lastly, the last column shows that 95 concepts have very high connectivity, being used in more than 10K ERs.

The distribution of concepts by ER is shown in Figure 2b. This distribution corresponds to the multi-valued property dct:subject. The first three columns show that the majority of ERs are associated with less than 300 concepts (less than 300 values for the property dct:subject for the same ER). On the other hand, the last three columns show that only a small number of ERs (40 ERs) are linked to a high number of concepts.

The pageRank score of a concept is local to an ER, it depends on the number of concepts that the Wikifier associates with this ER. The sum of the pageRank values of all concepts linked to an ER is 1. Thus, the greater the number of concepts, the lower their pageRank score.

3. Data transformation pipeline

Figure 3 shows the pipeline for our ETL process. All related files can be found in the pipeline repository[8]. The extraction phase involves collecting data from a Postgres database. The data is extracted as JSON files because one of the attributes of the Postgres database contains JSON data, thus converting everything to JSON was deemed more efficient. In the transformation phase, the JSON files are converted into semantic RDF triples. To compare the different RDF reification models, four RML mappings were created in order to obtain standard reification, singleton properties, named graphs, and RDF-star. In particular, RML-star [5] is used to generate RDF-star data. SHACL is then used to validate our RDF graphs. In the loading phase, 15 different docker containers were loaded and setup using docker-compose. Jena, GraphDB, and Oxigraph, have one KG instance by reification model. Virtuoso has only 3 KG instances because it does not supports RDF-star. The rest of this section focuses on explaining the four different reification models, and then it explains the RDF-star mappings.

3.1. Reification models

Standard reification. The standard reification model was proposed within RDF primer standardised by W3C [6][9]. In this model, rdf:Statement is used to define the triple that

[8] https://gitlab.univ-nantes.fr/clara/pipeline
[9] https://www.w3.org/TR/rdf-primer/#reification

```
uns:S100  rdf:type rdf:Statement ;
          rdf:subject unr:ER ;
          rdf:predicate dct:subject ;
          rdf:object :Query_Language ;
          uno:cosineSimilarity "0.6" ;
          uno:pageRank "0.4" .
```

(a) Standard reification in Turtle

```
unr:ER dct:subject :Query_Language <g-100> .
<g-100> uno:cosineSimilarity "0.6" ;
        uno:pageRank "0.4" .
```

(b) Named Graphs in N-quads

```
unr:ER   <p-200> :Query_Language .
<p-200>  rdf:singletonPropertyOf dct:subject;
         uno:cosineSimilarity "0.6" ;
         uno:pageRank "0.4" .
```

(c) Singleton properties in Turtle

```
<< unr:ER dct:subject :Query_Language >> uno:cosineSimilarity "0.6" ;
                                         uno:pageRank "0.4" .
```

(d) RDF-Star in Turtle.

Figure 4. Syntax of the four reification models.

will be annotated (rdf:subject, rdf:predicate, and rdf:object). The defined statement can be identified by a blank node or a URI. Figure 4(a) gives the representation in RDF triples. It displays two score values annotated on a statement.

Named graphs. Carroll et al. [7] proposed an extension to the RDF data model that allows RDF graphs to be named by URIs, which are referred to as named graphs. In this approach, a named graph is represented as a pair (g, n), where g is an RDF graph and n is an IRI, a blank node, or a default graph. The statements to be annotated are defined in one RDF graph, while the annotations themselves are defined in another RDF graph. The annotations are directly linked to this graph. Figure 4(b) shows the syntax for this example.

Singleton properties. The singleton property model [8] proposes creating a unique property for every triple that has associated metadata. In this model, a new node is created to represent the new property, which is connected directly to the original annotation property using the proposed property *singletonPropertyOf*. The same property is used for all metadata associated with a statement. An example of RDF triples using this model is shown in Figure 4(c).

RDF-Star. [9][10] proposed RDF-star and SPARQL-star as extensions to RDF and SPARQL to enable graph nesting and simplify the representation of reified statements. RDF-star and SPARQL-star allow for the recursive nesting of graphs, eliminating the need for declaring edge identifiers that are linked with metadata. RDF-star enables the nesting of triples within other triples as subjects or objects by using double angle brackets $\ll \gg$. As a result, every reified statement can be interpreted as a single RDF triple. An example of RDF-star reification is shown in Figure 4(d).

3.2. Mapping JSON to RDF

The aggregate of the JSON files used as input to generate the four versions of our KG can be seen in Figure 5. Each ER is described by an id, a title, a description, etc. It can have several authors and can be associated to several concepts.

We define four mappings, one for each reification model. We use the RML language [10] to generate the RDF triples for standard reification, singleton property and named graph. To generate RDF-star we use the RML-star [5] language. Listing 1 shows an excerpt of the RML-star mapping used to transform the JSON data into RDF-star. This

[10] https://w3c.github.io/rdf-star/cg-spec/2021-12-17.html

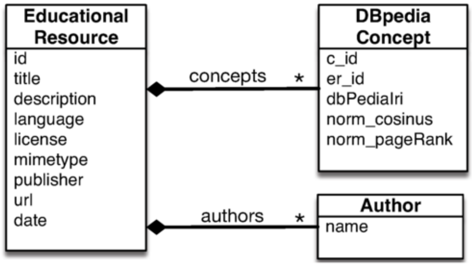


Figure 5. JSON aggregate of the source files for the RML mappings.

excerpt contains the RML-star rule that generates the triple with the reified property dct:subject (Lines 1 to 16) and the rule to generate one annotation (Lines 18 to 33). Lines 2 to 6 indicate an iteration over every ER in the JSON file. Lines 7 to 9 define the id of the ER as a subject for the triple. Lines 10 to 16 define the predicate for this subject (dct:subject) and multiple objects. These multiple objects are the set of concepts obtained from the JSON data. They are DBpedia resources treated as IRIs. Lines 19 to 23 again iterate over every ER in the JSON file. This time to generate the annotation. Lines 24 to 26 show that :ER_concept_link, that is the name of the first rule, is now the subject. Lines 27 to 33 define the annotation of the pageRank score, taken from the JSON under the attribute "norm_pageRank". In a similar way, the annotation for cosineSimilarity is generated.

We use Morph-KGC [11] as mapper to generate the RDF triples. Morph-KGC was able to generate the four versions of our KG through the four mappings including the RML-star mapping.

```
1   :ER_concept_link a rr:TriplesMap;
2       rml:logicalSource [
3           rml:source "json/ER/normal.json";
4           rml:referenceFormulation ql:JSONPath;
5           rml:iterator "$.resources[*]";
6       ];
7       rr:subjectMap [
8           rr:template "https://unknown.com/resource/{id}";
9       ];
10      rr:predicateObjectMap [
11          rr:predicate dct:subject;
12          rr:objectMap [
13              rml:reference "concepts.dbPediaIri";
14              rr:termType rr:IRI;
15          ]
16      ].
17
18  :ER_concept_context a rr:TriplesMap;
19      rml:logicalSource [
20          rml:source "json/ER/ER_0.json";
21          rml:referenceFormulation ql:JSONPath;
22          rml:iterator "$.resources[*]";
23      ];
24      rml:subjectMap [
25          rml:quotedTriplesMap :ER_concept_link;
26      ];
27      rr:predicateObjectMap [
```

```
        rr:predicate uno:pageRank;
        rr:objectMap [
            rml:reference "concepts.norm_pageRank";
            rr:datatype xsd:double;
        ]
    ].
```

Listing 1: Excerpt of the RML mapping for RDF-star reification.

4. Experimental evaluation of reification models

In our KG, the relation dct:subject is reified with two annotations (uno:pageRank and uno:cosineSimilarity). The number of values for this relation can be high (up to several thousands of concepts by ER, cf. Figure 2b) thus, the number of annotations by ER can be huge. The goal of this section is to compare four different reification models expressed over a multi-valued property. The triplestores we analyse are Virtuoso, Jena, GraphDB, and Oxigraph.

The rest of this section is organised as follows. First, Section 4.1 compares the four analysed reification models in terms of syntax and number of triples. Then, Section 4.2 describes the setup of our experiments. Section 4.3 shows the results of our experiments, comparing the size of the triplestores and the execution times of the different queries. Finally, Section 4.3 analyses the obtained results to extract our conclusion. All queries, the corresponding scripts, and the experiment results (raw execution times and plots) can be found in the queries_comparison repository[11].

4.1. Syntax comparison of analysed reification models

Described reification models differ in various criteria such as the total number of triples, flexibility, and syntax support.

Number of triples. Standard reification is the most costly approach since it needs five triples for each reified statement. Singleton properties needs three triples. Named graphs and RDF-star are the most compact models needing two and one triples respectively.

The second column of Table 1 shows the number of triples by reification model. In our KG, around 12M of triples are shared among all reification models. And more than 8M statements are reified. This amount of statements leads to the observed differences. As expected, RDF-star is the most compact model, followed by named graphs. The bulkiest model is standard reification followed by singleton properties.

Flexibility. All of these reification models are flexible when it comes to adding new annotations to an already reified statement. Adding new annotations only requires adding one additional triple for each approach. Additionally, all of these models cause no issues with multi-valued properties. Also, one advantage specific to named graphs is that reification can be defined also for a group of triples or even a dataset.

Syntax support. Standard reification and singleton properties conform to the core RDF model proposed in 2004. Named graphs represent an extension to the triple RDF model and is part of the standard RDF1.1, which was published in 2014. RDF-star proposes to extend the RDF specification further. Concerning the query language, all of these models are supported in the SPARQL standard, except for RDF-star which proposes SPARQL-star as a query language.

[11] https://gitlab.univ-nantes.fr/clara/queries_comparison

	Number of Statements	Virtuoso	Jena	GraphDB	Oxigraph
Standard reification	61,865,751	3.6 GB	52 GB	8.5 GB	6.5 GB
Singleton properties	45,335,637	3.6 GB	48 GB	257 GB	5.7 GB
Named graphs	37,071,104	3.2 GB	51 GB	6.4 GB	9 GB
RDF-Star	37,055,676	-	50 GB	6.6 GB	12 GB

Table 1. Generated DB size of different reification models.

The implementations of RDF-star[12] can follow three approaches: *PG (Property Graph), SA (Separate Assertion), or both*[13]. In the SA mode, quoted triples are not necessarily asserted in the graph. In the PG mode, any quoted triple is automatically asserted. Jena supports both modes, Oxigraph and GraphDB both support only the SA mode for RDF-Star. And Virtuoso does not support RDF-star at all.

4.2. Experimental setup

Experiments were run on a virtual machine with 128GB of RAM, 2GHz with 32 cores, on a Debian GNU/Linux 11 (Bullseye). All tests were run using docker images of the triplestores[14] [15] [16] [17]. All triplestores were parameterized with a query timeout of 30 minutes and given access to 16 GB of RAM. Only Oxigraph was not parameterized as we did not find the way to do it. GraphDB was also parameterized to use a context index when dealing with the named graphs version of the graph. All four triplestores were evaluated with the four reification models except for Virtuoso that does not support RDF-star, and GraphDB does not support singleton properties in an efficient way. This is because GraphDB makes the assumption that there will be only a small number of properties in the graph. This issue is described on the website of GraphDB[18]. Details on how exactly the experiments were run are given at the end of this section.

Query templates. Based on the series of queries A, B, and F used in [12], we define seven query templates that are presented as SPARQL-star queries for simplicity. The templates will be referred to as Q1 to Q7. All templates can be seen in Figure 6. Q1 and Q5 are grounded with instances of ERs while Q2, Q3, Q4, and Q6 are grounded with instances of concepts. Q7 is not grounded. In these queries, only the subjects are quoted. Q2, Q3, and Q4 are star-shaped queries.

- Q1 is a property path query. It returns the list of concepts and the associated hierarchy of categories (with skos:broader*) for a given ER.
- Q2 is a FILTER query that compares the annotations of a concept. Given a concept, it returns the associated ERs whose pageRank score is greater than the cosine similarity using the FILTER keyword.
- Q3 is a join query. It returns the set of ERs associated to three given concepts.
- Q4 is similar to Q3 but with a FILTER that specifies how the three pageRank scores must relate together.

[12] https://w3c.github.io/rdf-star/implementations.html
[13] https://w3c.github.io/rdf-star/cg-spec/editors_draft.html#sa-mode-and-pg-mode
[14] https://hub.docker.com/r/secoresearch/fuseki
[15] https://hub.docker.com/r/tenforce/virtuoso
[16] https://hub.docker.com/r/oxigraph/oxigraph
[17] https://hub.docker.com/r/khaller/graphdb-free
[18] https://graphdb.ontotext.com/documentation/10.0/devhub/rdf-sparql-star.html

Q1

```
SELECT ?concept ?categorie
WHERE {
  [ER] dct:subject ?concept .
  ?concept dct:subject/skos:broader* ?categorie .
}
```

Q2

```
SELECT ?er ?pr ?cosine
WHERE {
   << ?er dct:subject [concept] >> uno:pageRank ?pr ;
                                   uno:cosineSimilarity ?cosine .

   FILTER(?pr > ?cosine)
}
```

Q3

```
SELECT ?er ?pr1 ?pr2 ?pr3
WHERE {
   << ?er dct:subject [concept_1] >> uno:pageRank ?pr1 .
   << ?er dct:subject [concept_2] >> uno:pageRank ?pr2 .
   << ?er dct:subject [concept_3] >> uno:pageRank ?pr3 .
}
```

Q4

```
SELECT ?er ?pr1 ?pr2 ?pr3
WHERE {
   << ?er dct:subject [concept_1] >> uno:pageRank ?pr1 .
   << ?er dct:subject [concept_2] >> uno:pageRank ?pr2 .
   << ?er dct:subject [concept_3] >> uno:pageRank ?pr3 .

   FILTER( ?pr1 > ?pr2 && ?pr2 > ?pr3 )
}
```

Q5

```
SELECT ?title ?license ?format ?language ?publisher
?creator ?created ?description ?concept ?pr
WHERE {
  << [ER] dct:subject ?concept >> uno:pageRank ?pr .
  [ER]   dct:title ?title ;
         dct:license ?license ;
         dct:format ?format ;
         dct:language ?language ;
         dct:publisher ?publisher .

  OPTIONAL { [ER] dct:creator ?creator }
  OPTIONAL { [ER] dct:created ?created }
  OPTIONAL { [ER] dct:description ?description }
}
```

Q6

```
SELECT ?er ?pr
WHERE {
   { << ?er dct:subject [concept_1] >> uno:pageRank ?pr }
   UNION
   { << ?er dct:subject [concept_2] >> uno:pageRank ?pr }
   UNION
   { << ?er dct:subject [concept_3] >> uno:pageRank ?pr }
}
```

Q7

```
SELECT ?concept (COUNT(?er) as ?number_of_er)
WHERE {
   << ?er dct:subject ?concept >> uno:pageRank ?pr .

   FILTER(?pr > 0.01)
}
GROUP BY ?concept
```

Figure 6. Query templates

Q5 is join query that in addition uses the OPTIONAL clause. It returns all the information available for a given ER, the list of associated concepts, and the corresponding pageRank score.

Q6 is a UNION query that gets the ERs associated with one of 3 given concepts, using the UNION operator.

Q7 is a GROUP BY query that uses the COUNT operator. It returns the number of ERs by concept and it filters out the results lower than a pageRank threshold.

Groundings. Query templates are grounded with instances selected beforehand. We chose the groundings in order to evaluate the difference between multi-valued properties with few subjects or objects and multi-valued properties with a large number of subjects or objects. We use the multi-valued property dct:subject linking our ERs and their corresponding DBpedia concepts. This multi-valued property goes both ways, one subject to multiple objects, and one object from multiple subjects. We selected six ERs (the subject of the multi-valued property) and six concepts (the object of the multi-valued property). The first two ERs lead to a small number of multi-valued properties (3 and 7 objects), the next two have a medium number of multi-valued properties (108 and 270 objects), and the last two have a large number of multi-valued properties (1067 and 2053 objects). For DBpedia concepts, the first two concepts lead to a small number of multi-valued properties (3 and 13 subjects), the next two have a medium number of multi-valued properties (620 and 1123 subjects), and the last two have a large number of multi-valued properties (11486 and 21523 objects).

Methodology. Queries were instantiated with a number of groundings depending on the query as some queries need three different concepts (Q3, Q4, Q6). In that case only three grounding were given, each composed of three concepts. Instantiated queries were executed sequentially in increasing order of the size of the corresponding multi-valued property. Each query was executed 3 times sequentially. So, the capability of the triplestores to cache previous results had an important role. Queries were sent to the SPARQL endpoints with HTTP using a Python script.

4.3. Experimental results

This section first compares the size of the triplestores, then it explains the query execution times, and finishes with an analysis of obtained results.

Storage size

Columns 3 to 6 of Table 1 shows the size of the different triplestores with the different reification models. This table allows to see the differences in the storage size, but it also shows how efficiently the triplestores store each of the four reification models.

In general, Virtuoso uses the least amount of storage space and Jena uses the most. Storage costs do not change a lot across the reification models except for GraphDB that need 257 GB to store our KG with the singleton properties version. This observation was already put into light in [13]. The small difference of the storage volume by column (except for GraphDB using singleton properties) can be explained by the fact that an important part of the graph is common to all reification models.

Query execution results

A visualization of the execution times in seconds per query and triplestore is presented in Figures 7 and 8. Execution time is presented in a logarithmic scale. Each bar considers all corresponding groundings (except for Q1). The beginning of a bar indicates the fastest execution time and the top the largest. The black line is the average execution time and the yellow line is the median.

Figure 7(a) displays the results for Q1 which is a star property path query. Q1 is very challenging as it navigates multiple times (once for each concept linked to the grounded ER) through the dense hierarchy of DBpedia categories. Only the first four groundings were successfully executed. Queries with large groundings crashed. In general, Q1 has large execution times (sometimes even reaching 12 minutes for Oxigraph). We recall that GraphDB does not scale well using singleton properties so we were unable to experiment it with our KG. The overall observation is that Virtuoso outperforms for Q1 regardless of the reification model (with all averages being about 13 seconds), except for RDF-star, which is not supported.

Figure 7(b) displays the results for Q2 whose particularity is to compare two annotations of the same statements. The general observation is that Virtuoso outperforms the other triplestores except for named graphs where Oxigraph behaves very well. For standard reification, Virtuoso is the fastest (with an average of 0.16 seconds) followed by GraphDB (with an average of 0.38 seconds), then by Jena (with an average of 6.21

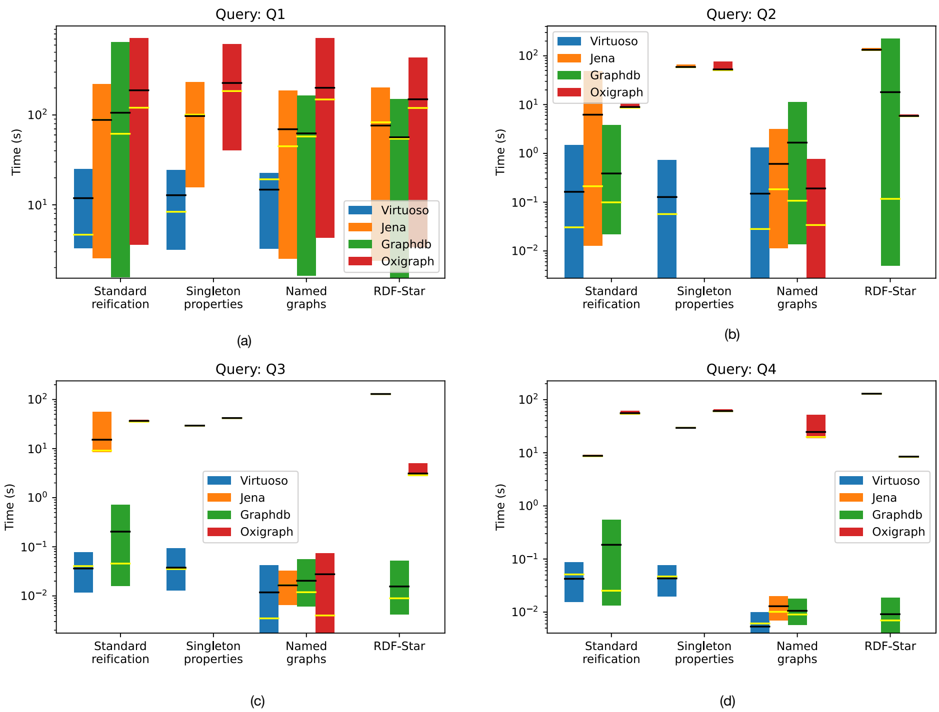


Figure 7. Comparison of Execution Time for Queries Q1-Q4 across Different Data Stores and Reification Approaches

seconds), and finally Oxigraph (with an average of 8.91 seconds). For singleton properties, like in Q1, only Virtuoso achieves good results. Jena and GraphDB have similar results (around 1 minute while Virtuoso achieves results on average less than 0.1 seconds). For named graphs, both Virtuoso and Oxigraph achieve similar results (around 0.1 second). They are followed by Jena, then by GraphDB. For RDF-star, once again GraphDB achieves good results followed by Oxigraph and Jena.

Figure 7(c) displays the results for Q3 that is a join query. Clearly, this figure shows that join queries are best executed over named graphs (all triplestores having an average of around 0.015 seconds). Concerning standard reification and singleton properties, Virtuoso outperforms the other three triplestores. For singleton properties again only Virtuoso achieves good results. For RDF-star, GraphDB achieves the best results (on average around 0.015 seconds). It is followed by Oxigraph, then far by Jena.

Figure 7(d) displays the results for Q4 that is a join query similar to Q3 but with a FILTER comparing the pageRank scores. Results are very similar to those of Q3. Execution times are higher than for Q3 but in general the cost of the FILTER is not very high, except for Oxigraph over named graphs. The average of Virtuoso, Jena, and GraphDB are around 0.01 seconds for named graphs, and Oxigraph's average is of 24.6 seconds. Again, GraphDB performs the best with RDF-star (in average 0.009 seconds).

Figure 8(a) displays the results for Q5. The singularity of this query is the use of the OPTIONAL clause. Again globally Virtuoso behaves the best (in average around 0.047

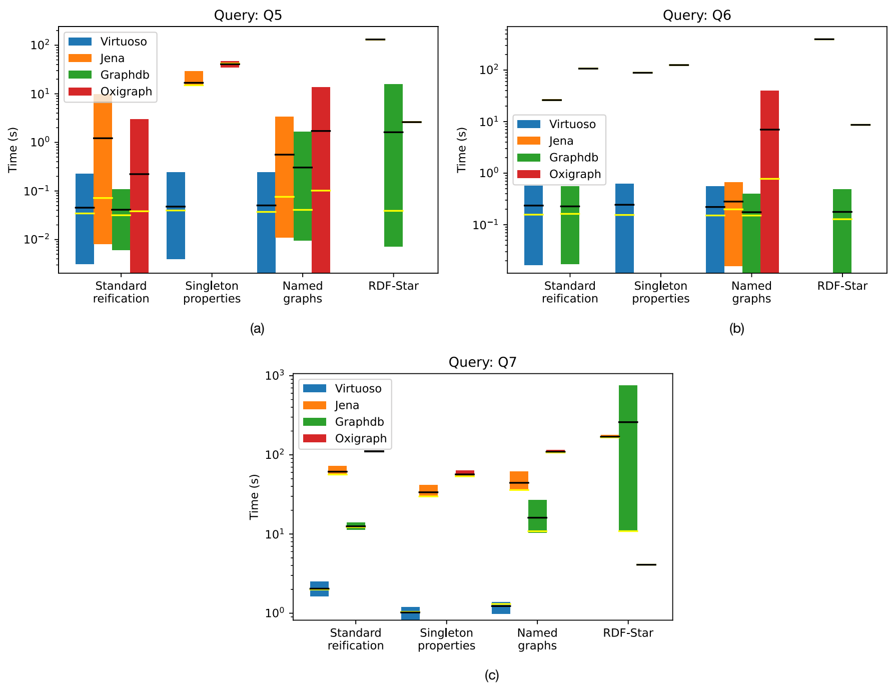


Figure 8. Comparison of Execution Time for Queries Q5-Q7 across Different Data Stores and Reification Approaches

seconds). For standard reification, GraphDB is as good as Virtuoso (with in average 0.041 seconds). GraphDB performs the best for RDF-star and Jena performs the worst.

Figure 8(b) displays the results of Q6. The particularity of this query is the use of the UNION clause. This query is globally well executed over named graphs (average of around 0.20 seconds) except for Oxigraph (in average of 6.99 seconds). In general GraphDB and Virtuoso behave the best. It is worth noting that again GraphDB executes well over RDF-star.

Figure 8(c) displays the results of Q7. The challenging aspect of this query is the GROUP BY operator. For that specific query, it is clear that Virtuoso has the best execution times regardless of the reification model (on average 1 or 2 seconds). The other triplestores are significantly slower (over 10 seconds), except Oxigraph with RDF-star (on average 4.09 seconds). Virtuoso is followed by GraphDB, then by Jena, and lastly by Oxigraph. It is worth noting that for Q7 Oxigraph behaves better than GraphDB on RDF-star.

Result analysis

The experimental evaluation done over our KG allows to draw the following conclusions. *Focus on reification models.* (a) Singleton properties is the least efficient reification model in our experiments. Only Virtuoso manages well this reification model. (b) Globally, standard reification and named graphs lead to good performances but named graphs is slightly better. (c) Even if in general RDF-star leads to bad execution times, frequently GraphDB obtains good results. (d) RDF-star is an elegant and compact model for statement-based annotations but triplestores should implement it more efficiently.

Focus on triplestores. (a) In general, Oxigraph is the least efficient triplestore in our experiments. However it is important to highlight that it performs better than Jena with RDF-star. (b) The execution times of Jena are consistently in second or third position but overall it outperforms Oxigraph. (c) In some experiments, GraphDB performs similarly to Virtuoso. (d) Virtuoso outperforms in most of our experiments but it should be noted that it does not support RDF-star.

The final conclusion for our KG is that the best choice would be Virtuoso with named graphs. Both standard reification and named graphs with Virtuoso exhibit similar performance. Named graph is slightly faster in some cases, in particular with join queries.

5. Related Works

Several works studied different reification methods and compared them according to several criteria. [13] focused on Wikidata and its representation in RDF using reification based on n-ary relations, standard reification, singleton properties and named graphs. Authors compared these models over five triplestores: 4store, Blazegraph, GraphDB, Jena, and Virtuoso. Their performance were measured based on 14 queries. Their results suggested that the singleton properties model was hardly supported but no other model was an outright winner. Concerning query performance, Virtuoso was the best followed by GraphDB and Blazegraph.

[14] realized an analysis of standard reification, named graphs, n-ary relations, singleton properties, companion properties (proposed in that paper) and RDF-star in its early stages. Experiments used Wikidata and DBpedia datasets on the triplestores Blazegraph, Stardog and Virtuoso. As DBpedia does not have singificant metadata, authors build a dataset with the Wikipedia revision history focusing on a company dataset. The experiments show that when the granularity of metadata is not by statement, companion properties and named graphs outperform. Concerning statement-level metadata, while standard reification results in the highest number of triples, it consumes the least storage in the database files and named graphs the most. This is because additional index structures for the graph identifiers are maintained. Concerning query performance, metadata characteristics have an impact on the reification models. Named graphs and RDF-star support queries against meta-metadata much better than the other models. In general, RDF-star can compete with named graphs if the metadata is on statement level. Moreover, both offer the best trade-off for mixed and data query workloads. In our experiments, queries do not contain data (i.e., triples) and we do not test querying meta-metadata.

[15] used three simple counting queries to analyse the internal representations of RDF-star in Stardog, Blazegraph and ExecuteSPARQLStar.[19] Experiments showed the divergence of the implementations of RDF-star when dealing with nested RDF-star statements. Blazegraph and ExecuteSPARQLStar behave similarly but Stardog was not able to deal correctly with nested RDF-star statements. That is because Stardog flattens the nested statements.

[16] proposed a data model called Labeled k-partite Graph (LKG) for storing and querying RDF triples with metadata. Authors compared experimentally LKG with Sin-

[19] https://github.com/RDFstar/RDFstarTools

gleton Property, RDF Reification, Named Graph and PaCE [17]. Used datasets were (with and without meta-knowledge): SPARQL Performance Benchmark (SP 2 Bench), the Biomedical Knowledge Repository (BKR), and the Gov-track. Results highlighed that LKG outperforms these methods by generating fewer statements, having a smaller graph size, avoiding resource redundancy, and achieving faster query response time.

[12] proposes a benchmark (dataset and set of queries) to analyse reification models. To illustrate the utility of the benchmark, authors analysed querying performance, storage efficiency and usability on the Stardog triplestore using three reification models: standard reification, singleton properties and RDF-star. Authors used the Biomedical Knowledge Repository (BKR) dataset[20] in order to make their results comparable with [8] that compares singleton property against standard reification. Twelve queries are used to evaluate performance. Five of these queries were proposed in this work to focus on SPARQL-star. Experimental results suggest that singleton property seems to have the worst performance. Probably because of the high number of unique properties and because indexes are usually not optimised with that in mind. Authors also observed that for simple queries, standard reification performs better than RDF-star. For complex queries, clearly, RDF-star outperforms standard reification.

[18] presents a novel approach for representing metadata, which outperforms existing reification models such as Singleton Property, Named Graph, PaCE, Companion Property, N-ary relations, RDF-star, and RDF-star [19]. The authors employed various datasets for their study, including a BKR dataset, a Gov-track dataset, a Synthetic dataset, and a dataset obtained from [20]. Through experiments, the proposed approach demonstrates advantages in handling multi-dimensional and nested metadata with reduced graph size and fewer generated statements.

6. Conclusion

This paper presented the pipeline for the generation of a knowledge graph (KG) of educational resources (ER) and the evaluation of several reification models with several triplestores. The objective was to identify the most suitable approach for this KG. To achieve this, we defined seven query templates instantiated in 26 grounded queries. Within the KG, reification was used in a multi-valued property to add two annotations whose range is between 0 and 1. Based on the insights derived from this experimental study, we were able to draw meaningful conclusions. Both, standard reification and named graphs with Virtuoso, exhibit similar performance. Named graphs show a slight advantage in some cases, in particular for join queries. RDF-star should be implemented more efficiently if quoted triples are included in RDF 1.2. Finally, for the KG presented in this paper, Virtuoso with named graphs, emerges as a good choice.

Acknowledgments

This work has received a French government support granted to the Labex Cominlabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01. Authors thank Master students in Computer Science of Nantes University for her participation in some aspects of this work.

[20] https://zenodo.org/record/3894746#.ZAtF0S_pNpQ

References

[1] Mihalcea R, Csomai A. Wikify! Linking documents to encyclopedic knowledge. In: Conference on Information and Knowledge Management (CIKM); 2007. doi:10.1145/1321440.1321475.

[2] Brank J, Leban G, Grobelnik M. Semantic annotation of documents based on wikipedia concepts. Informatica. 2018;42(1):23-32.

[3] Alexander K, Cyganiak R, Hausenblas M, Zhao J. Describing Linked Datasets. In: Workshop on Linked Data on the Web (LDOW); 2009. .

[4] Albertoni R, Browning D, Cox S, Gonzalez-Beltran A, Perego A, Winstanley P, et al.. Data catalog vocabulary (DCAT)-version 2; 2020. Available from: https://www.w3.org/TR/vocab-dcat-2/.

[5] Delva T, Arenas-Guerrero J, Iglesias-Molina A, Corcho O, Chaves-Fraga D, Dimou A. RML-star: A declarative mapping language for RDF-star generation. In: International Semantic Web Conference (ISWC) Posters, Demos and Industry tracks; 2021. p. 5.

[6] Manola F, Miller E, McBride B, et al. RDF primer. W3C recommendation. 2004. Available from: https://www.w3.org/TR/rdf-primer/.

[7] Carroll JJ, Bizer C, Hayes P, Stickler P. Named graphs. Journal of Web Semantics. 2005;3(4):247-67. doi:10.1016/j.websem.2005.09.001.

[8] Nguyen V, Bodenreider O, Sheth A. Don't like RDF reification? Making statements about statements using singleton property. In: International World Wide Web Conference (WWW); 2014. p. 759-70. doi:10.1145/2566486.2567973.

[9] Hartig O. Foundations of RDF* and SPARQL* (An alternative approach to statement-level metadata in RDF). In: International Workshop on Foundations of Data Management and the Web (AMW); 2017. .

[10] Dimou A, Vander Sande M, Colpaert P, Verborgh R, Mannens E, Van de Walle R. RML: A generic language for integrated RDF mappings of heterogeneous data. Workshop on Linked Data on the Web (LDOW). 2014;1184.

[11] Arenas-Guerrero J, Iglesias-Molina A, Chaves-Fraga D, Garijo D, Corcho O, Dimou A. Morph-KGC star: Declarative generation of RDF-star graphs from heterogeneous data; 2022.

[12] Orlandi F, Graux D, O'Sullivan D. Benchmarking RDF metadata representations: Reification, singleton property and RDF. In: International Conference on Semantic Computing (ICSC); 2021. p. 233-40. doi:10.1109/ICSC50631.2021.00049.

[13] Hernández D, Hogan A, Krötzsch M. Reifying RDF: What works well with Wikidata? International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS). 2015;1457:32-47.

[14] Frey J, Müller K, Hellmann S, Rahm E, Vidal ME. Evaluation of metadata representations in RDF stores. Semantic Web Journal (SWJ). 2017;10:205—229. doi:10.3233/SW-180307.

[15] Orlandi F, Graux D, O'Sullivan D. How many stars do you see in this constellation? In: European Semantic Web Conference (ESWC), poster demo; 2020. p. 175-80. doi:10.1007/978-3-030-62327-2_30.

[16] Sen S, Malta MC, Katoriya D, Dutta B, Dutta A. Labeled k-partite Graph for Statement Annotation in the Web of Data. In: 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). IEEE; 2020. p. 63-71. doi:10.1109/WIIAT50758.2020.00014.

[17] Sahoo SS, Bodenreider O, Hitzler P, Sheth A, Thirunarayan K. Provenance Context Entity (PaCE): Scalable provenance tracking for scientific RDF data. In: Scientific and Statistical Database Management: 22nd International Conference, SSDBM 2010, Heidelberg, Germany, June 30–July 2, 2010. Proceedings 22. Springer; 2010. p. 461-70. doi:10.1007/978-3-642-13818-8_32.

[18] Sen S, Katoriya D, Dutta A, Dutta B. RDFM: An alternative approach for representing, storing, and maintaining meta-knowledge in web of data. Expert Systems with Applications. 2021;179:115043. doi:10.1016/j.eswa.2021.115043.

[19] Schueler B, Sizov S, Staab S, Tran DT. Querying for meta knowledge. In: Proceedings of the 17th international conference on World Wide Web; 2008. p. 625-34. doi:10.1145/1367497.1367582.

[20] Fu G, Bolton E, Rosinach NQ, Furlong LI, Nguyen V, Sheth A, et al. Exposing provenance metadata using different RDF models. arXiv preprint arXiv:150902822. 2015. doi:10.48550/arXiv.1509.02822.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230009

BiPaSs: Further Investigation of Fast Pathfinding in Wikidata

Leon MARTIN [a,1]

[a] *Media Informatics, University of Bamberg, An der Weberei 5, 96047 Bamberg.*
ORCiD ID: Leon Martin https://orcid.org/0000-0002-6747-5524

Abstract. *Purpose:* A previous paper proposed a bidirectional A* search algorithm for quickly finding meaningful paths in Wikidata that leverages semantic distances between entities as part of the search heuristics. However, the work lacks an optimization of the algorithm's hyperparameters and an evaluation on a large dataset among others. The purpose of the present paper is to address these open points.
Methodology: Approaches aimed at enhancing the accuracy of the semantic distances are discussed. Furthermore, different options for constructing a dataset of dual-entity queries for pathfinding in Wikidata are explored. 20% of the compiled dataset are utilized to fine-tune the algorithm's hyperparameters using the Simple optimizer. The optimized configuration is subsequently evaluated against alternative configurations, including a baseline, using the remaining 80% of the dataset.
Findings: The additional consideration of entity descriptions increases the accuracy of the semantic distances. A dual-entity query dataset with 1,196 entity pairs is derived from the TREC 2007 Million Query Track dataset. The optimization yields the values 0.699/0.109/0.823 for the hyperparameters. This configuration achieves a higher coverage of the test set (79.2%) with few entity visits (24.7 on average) and moderate path lengths (4.4 on average). For reproducibility, the implementation called BiPaSs, the query dataset, and the benchmark results are provided.
Value: Web search engines reliably generate knowledge panels with summarizing information only in response to queries mentioning a single entity. This paper shows that quickly finding paths between unseen entities in Wikidata is feasible. Based on these paths, knowledge panels for dual-entity queries can be generated that provide an explanation of the mentioned entities' relationship, potentially satisfying the users' information need.

Keywords. knowledge graphs, pathfinding, hyperparameter optimization, Wikidata

1. Introduction

To satisfy the users' information need more quickly, the result pages of modern web search engines such as Google[2], Bing[3], and Startpage[4] feature a variety of components in addition to the standard ranking of search results. One prominent example are knowledge panels, which are typically located in the top right corner of the result pages. These

---

[1]Mail: leon.martin@uni-bamberg.de.
[2]https://www.google.com (accessed 2023/05/26)
[3]https://www.bing.com (accessed 2023/05/26)
[4]https://www.startpage.com (accessed 2023/05/26)

Table 1. A subset of the information displayed in the knowledge panel variants of Google, Bing, and Startpage when the two queries *European Union* and *Alan Turing* are issued individually. Information from third-party sources like weather services has been left out. The search engines were set to English and the searches were conducted on 2023/05/26.

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

box-shaped interface elements are populated with information from purpose-built knowledge bases, namely Knowledge Graphs (KGs) [1]. In a blog entry from 2012, Google unveiled their KG as a means of improving their search engine through three primary functions: the disambiguation of entities, the generation of summaries of entities, and the provision of links to associated entities. Especially the latter two functions contribute to the composition of the knowledge panel's content. Using the example of two queries, Table 1 shows the variety of information that the knowledge panel variants of the three web search engines comprise when a query mentioning a single entity is issued. Note how the types of the queried entities, which are in this case a political union and a human being, affect the information presented in the knowledge panels. The reasons for this are twofold: First, the employed KGs use different properties to describe entities of different categories, and second, the search engines rank the entity information differently.

The examples demonstrate that knowledge panels for single-entity queries, i.e., queries mentioning exactly one entity, offer useful information. However, the quality of the knowledge panels decreases when dual-entity queries, i.e., queries mentioning exactly two entities, are issued. For instance, given the query *European Union Alan Turing*[5], Google displays no knowledge panel at all while Bing presents the same knowledge panel as for the query *Alan Turing*. Startpage shows a knowledge panel with information about a UK student exchange program named after Turing. While Startpage's result is a good attempt, the authors of [2] argue that knowledge panels for dual-entity queries could explain the relationship between the two mentioned entities, thereby potentially satisfying the users' information need without requiring them to consult the ranked search entries. Especially when the two entities are semantically distant, the way they are connected via a path in a KG, can provide valuable information for the users. The task of entity relationship explanation, which is well-known in the knowledge discovery discipline [3], is defined in [4] as follows:

> Given a pair of entities e and e', provide an explanation, i.e., a textual description, supported by a KG, of how the pair of entities is related.

[5] The searches were conducted on 2023/05/26.

The authors of [2] interpreted this task as a pathfinding problem, where a path found between e and e' in a KG serves as a means of describing the entity relationship. Web search engines like those mentioned before allow users to issue arbitrary textual queries with entities from basically any domain. Accordingly, they used Wikidata [5], a large general-domain KG, for their work since more specialized KGs do not encompass this scope. Applied to the previous example, one useful path between the entities *European Union* and *Alan Turing* in Wikidata is[6]:

Q458 (European Union) −P530 (diplomatic relation)→ Q145 (United Kingdom) ←P27 (country of citizenship)− Q7251 (Alan Turing)

Note that properties in a KG can be interpreted in both directions. Therefore, the edges of a KG can be considered as bidirectional [3]. The vast size and generality of KGs like Wikidata pose several concrete problems for pathfinding [2]:

- Uninformed search algorithms like breadth-first search might not suffice for pathfinding.
- KG interfaces struggle to deliver all edges of a queried entity.
- Users desire meaningful entity relationships.

To tackle these problems, [2] proposes a bidirectional A* search algorithm [6] that considers the semantic distance between entities estimated via word embeddings of their labels to guide the search. The algorithm is parameterized with three hyperparameters α, β, and γ that weight the individual components of the employed cost function (s. Section 2). Despite the promising results, the paper leaves several open points: The algorithm's performance was evaluated based on only twelve hand-selected dual-entity queries using hand-selected configurations of the hyperparameters. Moreover, further tests (s. Section 3) with the original algorithm indicated that the entity labels alone do not yield accurate word embeddings and thus compromise the search. To tackle these open points, the present paper investigates the following research questions:

RQ1: How to estimate the semantic distances between entities more accurately?
RQ2: How to obtain realistic dual-entity queries for pathfinding in Wikidata?
RQ3: What is an optimized hyperparameter configuration for the algorithm?
RQ4: How does the optimized hyperparameter configuration perform (against other configurations) on a large dual-entity query dataset?

The remainder of the present paper is organized as follows: Section 2 investigates foundations and related work, including a thorough recapitulation of [2]. Building upon this, Section 3 focuses on answering RQ1 through RQ3. Section 4 discusses the implementation developed in the context of the present paper[7], before RQ4 is addressed in Section 5. Finally, Section 6 draws a conclusion and provides directions on future work.

[6] In this notation, the properties within the arrows (edges) connect the surrounding entities (nodes) in the respective direction. The strings with leading Q and P are Wikidata's proprietary IDs for entities and properties.

[7] The implementation and all resources required to reproduce the results are available in the GitHub repository at `https://github.com/uniba-mi/bipass-wikidata-pathfinder`, which is also indexed in the Software Heritage Project's archive (`https://archive.softwareheritage.org`; accessed 2023/05/26).

2. Foundations & Related Work

The Resource Description Framework (RDF) [7] represents a generic approach for expressing knowledge in the form of triples. In this framework, Internationalized Resource Identifiers (IRIs) are utilized as identifiers, which are a generalization of Uniform Resource Identifiers (URIs). Each triple consists of a subject, which can be an IRI or a blank node, a predicate, which is an IRI, and an object, which can be an IRI, a literal, or a blank node. The predicate denotes a property, which is a binary relation between the subject and the object expressing a statement. Although the definitions vary in the community [8], a set of RDF triples that signify real-world entities and their relationships based on a predefined ontology can be referred to as a KG. A KG can therefore be interpreted as a graph $G = (V, E)$, where V (the nodes) is the combined set of subjects and objects, and E (the edges) the instances of predicates. Regarding the pathfinding problem, only IRI nodes qualify as entities between which paths can be searched. Accordingly, a path between a pair of entities e and $e' \in V$ is a subgraph of G, where e and e' are IRIs and each node on the path is either another IRI or a blank node. KGs typically provide SPARQL Protocol and RDF Query Language (SPARQL) [9] interfaces for issuing queries.

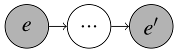
While there exist various domain-specific KGs like the Open Research Knowledge Graph (ORKG) [10] for the scientific domain, Wikidata represents a KG for the general domain. According to its statistics [11], Wikidata currently contains over 100 million entities that possess highly varied in- and outdegrees, where the indegree can exceed the outdegree by magnitudes. For example, issuing two simple SPARQL queries to the Wikidata Query Service[8] reveals that there are over 2,400 triples with the entity Q183 (Germany) as the subject and over 3.1 million with this entity as the object. Even though countries are entities with a particularly high in- and outdegree, there are numerous entities from other categories that have numbers in the thousands. The higher the indegree and outdegree of an entity, the higher the chances of encountering the entity when traversing the KG. As a result, uninformed search algorithms like breadth-first search do not suffice for fast pathfinding, in the worst case even if e and e' are directly adjacent nodes. Another problem is that the Wikidata Query Service already struggles to return mere 100,000 triples. Retrieving all adjacent entities connected to an entity via its incoming and outgoing edges is therefore not always possible.

2.1. An Algorithm for Fast Pathfinding in Wikidata

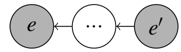
To mitigate the problems described above, the bidirectional A* search algorithm proposed in [2] only considers outgoing edges when traversing the KG and performs two simultaneous searches, one from e to e' and one from e' to e. This way, only paths of the patterns shown in Figure 1 can be discovered by their algorithm. In [2], the authors assume that this limitation is not problematic, at least in the context of Wikidata, due to its dense connectivity. Section 5 will assess this assumption.

Algorithm 1 shows a more verbose version of the strongly condensed algorithm as presented in [2]. The algorithm operates as follows: To keep track of the most promising entities to pursue during graph traversal, a priority queue is employed, which is initialized with e and e'. In each iteration, the first entity is taken from the queue. If a path

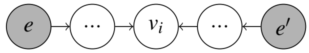
[8] https://query.wikidata.org (accessed 2023/05/26); all SPARQL query results mentioned in the present paper have been retrieved from the Wikidata Query Service on the date of visit.



(a) A direct path from e to e'



(b) A direct path from e' to e



(c) A path composed of a direct path from e to an intersecting entity v_i and a direct path from e' to an intersecting entity v_i

Figure 1. The graph patterns that can be found using the bidirectional A* search algorithm of [2]. e and e' denote the entities of a dual-entity query, between which a path was searched. Nodes with ... are placeholders for series of $n \geq 0$ entities.

Algorithm 1 The bidirectional A* search algorithm from [2]; more verbose and with adapted notation.

procedure FINDPATH(e, e', α, β, γ, *entityLimit*)
  $priorityQueue \leftarrow \langle e, e' \rangle$
  $reachable_{source} \leftarrow \{e\}$
  $reachable_{target} \leftarrow \{e'\}$
  $visitedEntities \leftarrow \{\}$ /
  while $priorityQueue = \emptyset$ **and** $|visitedEntities| < entityLimit$ **do**
    $entity \leftarrow dequeue(priorityQueue)$
    $visitedEntities \leftarrow visitedEntities \cup \{entity\}$
    if $entity \in (reachable_{source} \cap reachable_{target})$ **then**
      return $reconstructPath(e, e'), |visitedEntities|$
    end if
    for $adjacentEntity \in getAdjacentEntities(entity)$ **do**
      $costs \leftarrow calculateCosts(e, e', adjacentEntity, \alpha, \beta, \gamma)$
      $enqueue(priorityQueue, adjacentEntity, costs)$
      if $entity \in reachable_{source}$ **then**
        $reachable_{source} \leftarrow reachable_{source} \cup \{adjacentEntity\}$
      else if $entity \in reachable_{target}$ **then**
        $reachable_{target} \leftarrow reachable_{target} \cup \{adjacentEntity\}$
      end if
    end for
  end while
  return $\bot, |visitedEntities|$
end procedure

between e and e' can be established through the currently visited entity, the algorithm terminates, returning the found path and the number of visited entities. Otherwise, the adjacent entities are retrieved and enqueued with respect to the costs of the paths leading to them. The costs are calculated by means of a cost function with the hyperparameters α, β, and γ. When the priority queue becomes empty or the entity limit, i.e., the maximum number of entities that are allowed to be visited before the search is aborted, is reached without a path being found, the algorithm terminates unsuccessfully.

By providing the measure for ranking the entities in the priority queue, the cost function (*calculateCosts* in Algorithm 1) guides the graph traversal and ultimately determines the algorithm's performance as well as the characteristics of the found paths. What remains to be discussed is therefore what cost function can meet the requirement that users desire meaningful entity relationships in the knowledge panel context, as pointed out in

Section 1. Particularly in Wikidata, numerous paths between two entities can be found, thus posing the question which candidate path is the most meaningful. Since meaningfulness is a highly complex and subjective concept, there are different approaches tackling this problem from different directions. For example, [3] investigated how *informative* subgraphs explaining the relationship between entities can be mined from entity relationship graphs. To this end, the proposed approach ranks candidate nodes according to their informativeness, which is computed using edge weights that are based on co-occurrence statistics for entities and relationships. While this statistical approach yields subgraphs that are structurally important with respect to the query entities, it disregards available semantic information like the entity labels. Thus, the found subgraphs might be prone to concept drift [12], which occurs when the semantic focus of the query is left.

In comparison, the authors of [2] argue that a path with minimal concept drift is a meaningful path. Further, they propose to assess whether a certain entity is out of a query's semantic focus by means of the semantic distances between entities. To calculate the semantic distance they use the cosine distance between the fastText[9] word embeddings [13] of the entities' labels. fastText is a well-known library by Facebook that produces static vector representations for words, while being robust against misspelling. With respect to the general cost function $f(p) = g(p) + h(p)$ of the A* search algorithm [6], [2] introduces a cost function that leverages semantic distances and the path length to calculate the costs of a path p comprising n entities as follows[10] [2]:

$$
\begin{aligned}
&g(p) \coloneqq \alpha \cdot \overline{d}(p_{[..n-1]}, e') + \beta \cdot n \\
&h(p) \coloneqq \gamma \cdot d(v_n, e') \\
&\text{where } p = \langle e, \ldots, v_n \rangle \\
&\text{and } p_{[..f]} \text{ is the sub-path } \langle e, \ldots, v_f \rangle \text{ of } p
\end{aligned}
$$

Representing the first part of formula $g(p)$, the formula $\overline{d}(p_{[..n-1]}, e')$ calculates the average of the semantic distances between all entities on the path except the last and e'. The second part is supposed to add the path length to the costs[11]. The formula $h(p)$ estimates the costs of the remaining path by means of formula $d(v_n, e')$ as the semantic distance between the last entity on the path and e'. As shown, the cost function is parameterized with three hyperparameters α, β, and γ that weight its components. To calculate the costs of a converse path in the bidirectional search, i.e., a path that starts at e' with the goal of finding a path to e, e and e' are simply interchanged in the cost function.

In addition to preferring semantically meaningful paths, the usage of this cost function as a search heuristics serves a second purpose. As explained in [2], the semantic distance of an entity to other entities tends to positively correlate with the minimal number of hops between the entities. For instance, entities that are at least five hops apart typically have a higher semantic distance than entities that are two hops apart. By prioritizing entities with a lower semantic distance to a target entity, the chances of reaching the target entity earlier are therefore higher compared to a breadth-first search.

[9] `https://fasttext.cc` (accessed 2023/05/26)

[10] The cost function was slightly modified to comply with the notation introduced above.

[11] In [2], n was used instead of $n-1$ in the second part of $g(p)$, which is a mistake because the length of a path is typically defined as the number of edges, i.e., one less than the number of nodes on the path [14]. However, this mistake does not compromise the order in the priority queue since it applies to all paths equally.

Table 2. The four configurations for the hyperparameters α, β, and γ used in [2].

Name	α	β	γ	Description
Uninformed	0	1	0	Only considers the path length, similar to breadth-first search.
Semantics-Only	1	0	1	Ignores the path length and only considers the semantic distances.
Greedy	0	0	1	Estimates the total path costs as the semantic distance of the last entity on the path to e/e' depending on the search direction.
Balanced	1	0.5	1	Leverages all components of the cost function in a balanced setting.

In recent years, novel transformer-based approaches like BERT [15] yielded excellent results, thus replacing previous approaches including fastText as the state-of-the-art for various natural language processing tasks. There are numerous variants of BERT tailored to specific domains and tasks. One example is Sentence-BERT or SBERT [16], which is able to efficiently compute accurate vector representations of sentences. For SBERT, various pre-trained models are available, which are also tailored to specific tasks. Section 3 introduces how entity descriptions, which are often one or multiple sentences, can be leveraged in addition to entity labels to improve the accuracy of the semantic distances. One potent model fine-tuned for sentences as well as short paragraphs is `all-mpnet-base-v2`[12], which is based on Microsoft's MPNet [17]. Due to the characteristics of the new input and its general performance, fastText is replaced by SBERT in combination with `all-mpnet-base-v2` for the implementation of the cost function.

2.2. Hyperparameter Optimization

In [2], the authors evaluated the performance of the pathfinding algorithm using only the four intuitively set hyperparameter configurations shown in Table 2, which leaves room for improvement. The recent attention on machine learning fueled the investigation of hyperparameter optimization techniques, i.e., methods for automatically setting hyperparameters of objective functions to optimize performance [18]. One example is Bayesian optimization, a state-of the-art optimization framework for the global optimization of expensive blackbox functions that is applicable for a wide range of problems [19]. The Bayesian optimization framework can be broken down into two primary components [19,18]. Firstly, there is a probabilistic surrogate model that incorporates a prior distribution representing the beliefs about the unknown objective function's behavior. Secondly, an acquisition function, that measures the optimality of a series of queries, is utilized. The goal is to minimize the anticipated loss to determine the optimal sequence of queries. Based on the output of each query, the prior is revised, resulting in an informative posterior distribution over the objective function's space. Due to this incremental approach to optimization, it outperforms basic hyperparameter optimization techniques like grid search both in terms of the hyperparameter quality and efficiency.

A related alternative is called Simple(x) or just Simple [20]. While Bayesian optimization uses computationally expensive Gaussian processes to model the objective function, Simple creates a model by dividing the optimization area into simplices. The algorithm iteratively tests points within each simplex to create a more precise model. Thereby, this approach converts the optimization task into a dynamic programming problem, allowing samples to be taken without updating the entire model. Hence, Simple is employed for the optimization of α, β, and γ in Section 3.

[12] `https://huggingface.co/sentence-transformers` (accessed 2023/05/26)

3. Algorithm Improvements and a Query Dataset

Based on the insights from Section 2, this section aims to answer the research questions RQ1 to RQ3. Since RQ4 is related to the evaluation, its discussion follows in Section 5.

3.1. Accurate Semantic Distances between Entities

As described in Section 2, [2] estimates the semantic distance between two entities as the cosine distance between the vector representations of the entity labels. However, the ambiguity of entity labels alone compromises the accuracy of the vector representations and therefore the resulting semantic distances. For example, Wikidata features numerous entities that share the label *Paris*, partly from very different categories: While the entity Q90 refers to the city in France, the entity Q167646 refers to the mythological son of Priam, king of Troy[13]. During pathfinding, it is important to pin down the exact entity that is currently examined. Otherwise, paths with concept drift might be pursued when entities with alleged low semantic distances are prioritized. Therefore, RQ1 asks how the semantic distances between entities can be estimated more accurately.

We propose the two following changes to mitigate this problem. The first change is to feed entity descriptions, another resource that is available in Wikidata for most entities, in addition to entity labels to the word embedding model. For this purpose, the entity labels and entity descriptions are simply concatenated. The idea is that the additional information provided by the descriptions results in vector representations that capture the entities' meaning more accurately. Secondly, fastText is replaced by SBERT in combination with `all-mpnet-base-v2` for efficiently computing high-quality vector representations of the new input type, i.e., strings composed of entity labels and descriptions. Also note that the data within Wikidata is curated. Hence, encountering misspelling is unlikely such that fastText's robustness against them is not required.

The examples in Table 3 show the positive impact of these changes on the semantic distances. In particular, the first example shows that the additional consideration of entity descriptions affirms the semantic distance in cases where the labels alone are expressive enough to compute accurate vector representations. In contrast, the pairs of examples two/three and four/five demonstrate how the entity descriptions help to disambiguate entities with identical or similar labels, yielding more accurate semantic distances.

Further qualitative experiments with other entity pairs conform with these observations. Hence, we conclude that the additional consideration of entity descriptions improves the accuracy of the semantic distances, thereby answering RQ1. That being said, apart from entity descriptions, Wikidata provides even more entity-related information including alternative labels, labels in other languages etc. Furthermore, the (direct) neighborhood of an entity can also be seen as a description of its meaning. To retain the focus of this paper, though, these options are left open for future work as their exploration seems worthwhile to further improve the accuracy of the vector representations.

3.2. A Dual-Entity Query Dataset for Pathfinding in Wikidata

Even though the twelve dual-entity queries discussed in [2] suffice for showcasing the potential of the algorithm, their low number and artificial hand-selected nature does nei-

[13] `https://www.wikidata.org/wiki/{Q90|Q167646}` (accessed 2023/05/26)

Table 3. Comparison of semantic distances using five examples with two Wikidata entities each. d_{labels} and $d_{labels+descs}$ denote whether the presented semantic distances were calculated using SBERT vector representations of the entity labels alone or of the concatenated entity labels and entity descriptions.

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

ther allow for a proper evaluation nor hyperparameter optimization. Accordingly, RQ2 raises the question what an appropriate dual-entity query dataset for pathfinding in Wikidata is. Considering the long-term goal of applying the algorithm for the knowledge panel generation in web search engines, such a query dataset has to be realistic in the sense that the queries have to be derived from queries issued to web search engines by actual users. However, no such query dataset (or benchmark) has been proposed so far.

In information retrieval research, the TREC [21] datasets are particularly popular and have been used for the evaluation of various information retrieval systems. The datasets are designed to enable researchers to evaluate the performance of their information retrieval systems using a common set of test collections. TREC has produced many different datasets over the years, covering a range of domains and types of text. At first glance, the datasets from the Entity Track[14] of TREC 2009, 2010, and 2011 appear to be useful for deriving a dual-entity query dataset for pathfinding in Wikidata because they include collections of entities for the evaluation of entity-oriented search systems. However, the derivation of a dual-entity query dataset from these datasets would require the artificial pair-wise combination of the single entities within the collections, which clearly contradicts the realism requirement. Furthermore, the TREC datasets only provide ClueWeb09[15] IRIs as identifiers for the entities, which would have to be expensively linked to Wikidata entities first.

Hence, another approach was pursued. The dataset from the Million Query Track[16] of TREC 2007 consists of 10,000 realistic textual queries for web search engines. Using this as a starting point, the following procedure was applied to each query of the TREC dataset to derive a dual-entity query dataset for pathfinding in Wikidata:

[14] https://trec.nist.gov/data/entity.html (accessed 2023/05/26)
[15] https://lemurproject.org/clueweb09/index.php (accessed 2023/05/26)
[16] https://trec.nist.gov/data/million.query07.html (accessed 2023/05/26)

1. GENRE[17] [22], a state-of-the-art Wikidata entity linker, is employed to look for Wikidata entities in the query. If two or more entities are recognized, the next step is taken. Otherwise, the query is dropped.
2. Since GENRE only provides the entity labels and not the necessary entity IDs, Wikidata is queried using SPARQL to retrieve them. To minimize incorrect matches[18], only entities with the exact labels are retained. If two or more entities are identified, the next step is taken. Otherwise, the query is dropped.
3. Finally, $\frac{n(n-1)}{2}$ pairs of entities are composed, where n is the number of recognized and successfully identified entities. The two IDs of each entity pair represent one dual-entity query, which is finally stored.

As an example, consider the query *children books on the effect of music on plants* with number 4480 from the TREC dataset. In the first step, GENRE recognizes three entities within this query, namely entities with the labels *book*, *music*, and *plant*. Querying Wikidata in the second step identifies the entities as Q571, Q638, and Q756, which are reasonable matches[19]. In the final step, $\frac{3(3-1)}{2} = 3$ dual-entity queries are composed, namely $\langle$Q571, Q638$\rangle$, $\langle$Q571, Q756$\rangle$, and finally $\langle$Q638, Q756$\rangle$. Note how the queries correctly reflect the order of occurrence of the entities in the original TREC query. In total, 1,196 dual-entity queries are derived using this procedure[20].

3.3. Optimization of the Hyperparameters α, β, and γ

The dual-entity query dataset not only allows for a proper evaluation of the pathfinding algorithm but also the optimization of its hyperparameters. For the reasons explained in Section 2, the Simple optimizer is leveraged to find an optimized hyperparameter configuration for the pathfinding algorithm, which represents the answer to RQ3. For this purpose, the optimizer requires an objective function that accepts a hyperparameter configuration and returns some objective value to assess the performance of this configuration. Depending on the use case, the optimizer's task is then to either minimize or maximize the objective value by testing different configurations.

Given the web search engine context, the goal is to find paths between entities in Wikidata fast, i.e., with few visited entities, such that the result can be displayed quickly. Hence, the number of visited entities is chosen as the objective value to be minimized. To account for the low-latency requirement of web search engines, the entity limit is set to 100. Furthermore, the pathfinding algorithm must be able to reliably find paths for unseen dual-entity queries because users are allowed to enter arbitrary queries. To evaluate this ability, the optimization operates on only 20% or 239 of the dual-entity queries while the remaining 80% or 957 queries serve as the test set for the evaluation in Section 5. Since the queries within the TREC 2007 Million Query Track dataset are ordered arbitrarily, the first 239 dual-entity queries are sampled for the optimization.

[17] `https://github.com/facebookresearch/GENRE` (accessed 2023/05/26)

[18] GENRE cannot always link entities correctly since it relies on the context of the entity mentions [22], which is sparse in many TREC queries. As explained before, entity labels alone are also ambiguous.

[19] Instead of Q571, a better match might have been Q8275050 (children's book), which demonstrates that entity linking is still an open problem; cf. `https://www.wikidata.org/wiki/{Q571|Q638|Q756|Q8275050}` (accessed 2023/05/26).

[20] The derived dual-entity query dataset is available in the provided GitHub repository[7] in the CSV format.

Algorithm 2 The algorithm for optimizing the hyperparameters α, β, and γ.

Require: A list *samples* with 239 dual-entity queries, the pathfinding procedure FINDPATH, the *minimize* function of the Simple optimizer

```
procedure TESTCONF(α, β, γ)
    score ← 0 // lower is better
    entityLimit ← 100
    for ⟨e, e'⟩ ∈ samples do
        path, numberOfVisitedEntities ← FINDPATH(e, e', α, β, γ, entityLimit)
        if path == ⊥ then
            score ← score + numberOfVisitedEntities × 2
        else
            score ← score + numberOfVisitedEntities
        end if
    end for
    objectiveValue ← score / |samples|
    return objectiveValue
end procedure

procedure PERFORMOPTIMIZATION
    interval ← [0.0; 1.0]
    iterations ← 150
    minValue, α, β, γ ← minimize(TESTCONF, interval, iterations)
    return minValue, α, β, γ
end procedure
```

In the literature on Bayesian optimization, the recommendations for the number of iterations vary depending on factors like the number of hyperparameters and the employed acquisition function (cf. [23]). [20] reports that Simple approximates the global optimum of an objective function with two hyperparameters at about 25 iterations. Thus, generous 150 iterations are used for our problem with three hyperparameters. Since the hyperparameters α, β, and γ represent weights of the cost function components, specifying that each of them can assume a value in the interval [0.0;1.0] suffices. With respect to these settings, Algorithm 2 presents the PERFORMOPTIMIZATION procedure for optimizing the hyperparameters α, β, and γ. To this end, the procedure calls the Simple optimizer's *minimize* function to minimize the *objectiveValue* returned by the TESTCONF procedure. Hence representing the objective function, TESTCONF accepts a candidate hyperparameter configuration and returns the average number of entities visited during pathfinding across all queries from *samples* using this configuration as the *objectiveValue*. If no path is found before the entity limit is reached, the number of visited entities is doubled as a penalty[21] for the particular query.

Figure 2 shows the *objectiveValue* yielded in each iteration of the optimization process as well as the *minValue*, i.e., the so far smallest *objectiveValue*. As depicted, a good *minValue* is already found in the 17th iteration and no major improvements are observed until the 150th iteration. Significant improvements beyond 150 iterations are therefore not expected. The lowest *minValue*, i.e., 59.347, was encountered in the 101st iteration using the hyperparameter configuration

$$\alpha = 0.699,\ \beta = 0.109,\ \gamma = 0.823$$

[21] Tests with other penalty factors and high fixed values as a penalty did not yield better results.

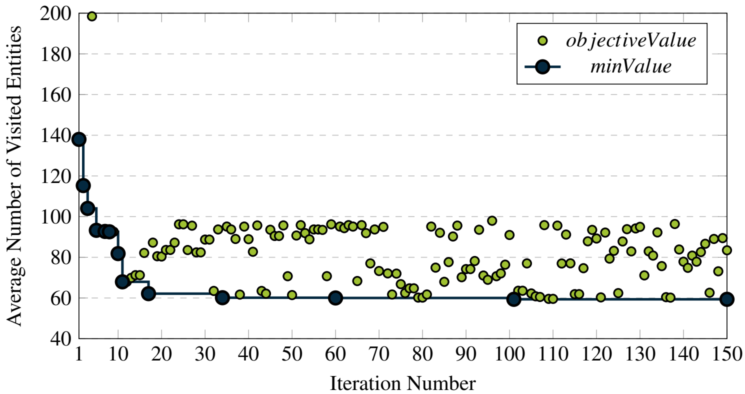


Figure 2. The optimization results.

which represents the optimized hyperparameter configuration and thus the answer to RQ3. Interestingly, β is significantly lower than the other hyperparameters. This supports the assumption that leveraging semantic distances as part of the search heuristics can reduce the number of visited entities. In the course of answering RQ4, Section 5 elaborates on this point.

4. The Implementation: BiPaSs

Another point that needs to be addressed even though it is not directly related to the answering of the research questions is the implementation. Originally, the prototype implementing the pathfinding algorithm was written in Python[22]. For the present paper, a full re-implementation[7] was produced. For future reference, we call the implementation Bidirectional Pathfinding System (BiPaSs). The new implementation uses Rust[23] for the pathfinding algorithm itself and leverages state-of-the-art data structures including a Fibonacci-heap-based priority queue. As shown in Figure 3, the implementation comprises four components. The first one is the *Query Factory* that implements the procedure from Section 3 for deriving the dual-entity query dataset from the TREC dataset. The resulting dual-entity queries are provided to the *Pathfinder* component, which contains the pathfinding algorithm as well as the code for running the hyperparameter optimization and the benchmark. For calculating the semantic distances using SBERT and `all-mpnet-base-v2`, it interacts with the *Wembed API* component via HTTP. For retrieving entity data from Wikidata, it interacts with the *Wikidata API* component, which is a wrapper for issuing SPARQL queries to the Wikidata Query Service, also via HTTP. The *Pathfinder*, the *Wikidata API*, and the *Wembed API* thus constitute the pathfinding system. For ease of use and reproducibility, all components are Docker-ized[24]. Additionally, a significant number of the HTTP interactions are cached, allowing for rapid reproduction of the results despite the restrictive query limits of the Wikidata Query Service, which are responsible for the major portion of the pathfinding duration.

[22] `https://www.python.org` (accessed 2023/05/26)
[23] `https://www.rust-lang.org` (accessed 2023/05/26)
[24] `https://www.docker.com` (accessed 2023/05/26)

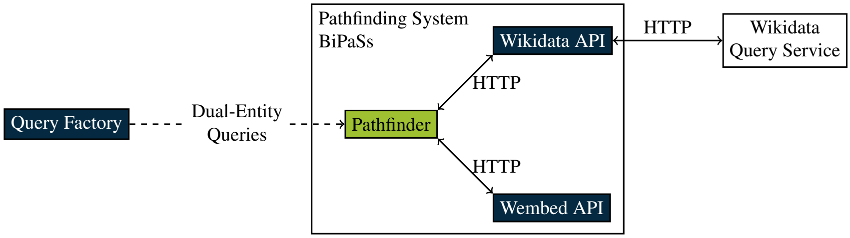


Figure 3. The components of the implementation. Components with a blue background are implemented in Python, the components with a green background in Rust. Arrows indicate communication between components. If the line is dashed, the communication takes place before and not during the actual pathfinding.

5. Evaluation & Discussion

To answer RQ4, the optimized hyperparameter configuration is benchmarked against all hyperparameter configurations introduced in [2] (s. Table 2). This includes the Uninformed configuration with α, β, and γ set to 0, 1, and 0 respectively that represents a baseline as it mimics bidirectional breadth-first search. For the pathfinding itself, the entity limit of 100 introduced in Section 3 is retained. As the test set of queries, the remaining 80% of the dual-entity query dataset, i.e., 957 unseen queries, are employed.

The first quantitative metric to be discussed is the coverage of the test set, i.e., the number of queries, for which a path was found, divided by the total number of queries. As shown in Figure 4, all configurations that consider semantic distances result in a higher coverage than the Uninformed configuration at only 55.6%. Due to the imposed entity limit, this supports the assumption that the usage of semantic distances as part of the search heuristics increases the chances of finding a path with fewer visited entities. However, the numbers of the other configurations show a considerable spread: While the Greedy configuration (59.4%) barely surpasses the Uninformed configuration, the Optimized configuration results in a coverage of 79.2%, the highest coverage achieved. Therefore, the optimization can be considered successful even though the Semantics-Only configuration from [2] actually turned out to be a strong guess as it results in a coverage of 73.6%. Raising the entity limit would increase the coverage of all configurations but also the pathfinding duration, which is problematic in the web search engine context.

Next, Figure 4 also reveals that the Optimized configuration visits 24.7 entities on average to find paths in the successful cases[25]. With 23.4, only the Semantics-Only configuration beats this. The configurations with a higher β, i.e., Balanced and Uninformed, need to visit the most entities to successfully find paths on average (30.0 and 39.5 respectively), which explains their lower coverage. The Optimized configuration also features a β value higher than 0, though. This indicates that β can affect the pathfinding positively up to a certain threshold, beyond which its impact becomes negative.

As the final metric, Figure 4 shows the average length of the paths found using the five hyperparameter configurations, again only considering the successful cases[25]. There are two groups of configurations. The first group comprises the Optimized, the Balanced, and the Uninformed configuration that produce paths with lengths of less than five on

[25] Only the numbers for successfully found paths are considered here due to the imposed entity limit. Otherwise, the entity limit would skew the results because it is not known whether a path would have been found using a certain configuration after visiting, for example, 105 or 1,005 entities.

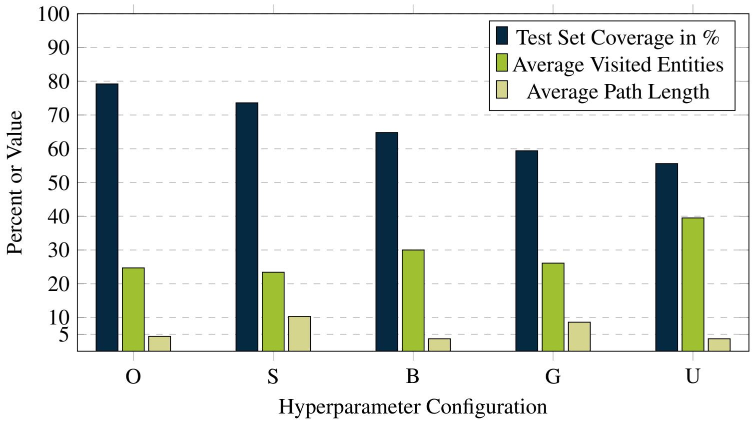


Figure 4. The benchmark results in terms of the coverage of the test set ($n = 957$), the average number of visited entities, and the average path length. The characters on the y-axis denote the examined hyperparameter configurations: **O**ptimized, **S**emantics-Only, **B**alanced, **G**reedy, and **U**ninformed.

average. The second group, i.e., the Semantics-Only and the Greedy configuration, yield average path lengths about twice as high. Note that the members of the former group use a β value of higher than 0, whereas the members of the latter use a β value of exactly 0. This indicates that β plays a key role in controlling the path lengths.

In summary, the answer to RQ4 is as follows: The Uninformed configuration reaches a lower coverage than the configurations that consider semantic distances, thereby supporting their utility as part of the search heuristics. Given the web search engine context, the Optimized configuration represents the best option as it reaches the highest coverage of the test set with a low average of visited entities and a moderate average path length. Its high coverage also supports the assumption by [2] that limiting the algorithm to only consider outgoing edges is unproblematic. At first, the Semantics-Only configuration seems to be a competitive configuration, as well. However, the high average length of the paths found with this configuration raises doubts about their usefulness for users.

To investigate the usefulness of entity relationships, a representative user study has to be conducted in the future. The goal of this study is to deepen the understanding of the meaningfulness of entity relationships in the web search engine context and to assess to what extent the cost function and hyperparameter configurations comply with the users' perceived meaningfulness of entity relationships. Nevertheless, to give an idea of the paths found by the different configurations, Table 4 presents a few examples. Generally, the examples conform with the quantitative results. One interesting observation is that the Semantics-Only path for Query A exhibits concept drift even though the employed configuration fully depends on semantic distances. Also, the Balanced and Uninformed configurations fail to find paths for Query B given the entity limit. The Optimized configuration produces adequately long paths that retain the semantic focus of the queries.

6. Conclusion

With the goal of investigating the open points of [2], the key contributions of the present paper include the improvement of the semantic distances leveraged in the pathfinding

Table 4. Examples of paths found using the five hyperparameter configurations. For conciseness, the predicates have been left out.

Configuration	Path found for Query A ⟨Q7958 (explanation), Q46857 (scientific method)⟩
Optimized	Q7958 (explanation) → Q352842 (teaching) → Q11862829 (academic discipline) ← Q336 (science) ← Q46857 (scientific method)
Semantics-Only	Q7958 (explanation) → Q352842 (teaching) → Q133500 (learning) → Q14819853 (learning or memory) → Q2996394 (biological process) → Q64732777 (biological phenomenon) → Q420 (biology) → Q7205 (paleontology) → Q1069 (geology) → Q7991 (natural science) → Q2522419 (hard science) → Q336 (science) ← Q46857 (scientific method)
Balanced	Q7958 (explanation) → Q352842 (teaching) → Q11862829 (academic discipline) ← Q336 (science) ← Q46857 (scientific method)
Greedy	Q7958 (explanation) → Q352842 (teaching) → Q11862829 (academic discipline) → Q336 (science) → Q46857 (scientific method)
Uninformed	Q7958 (explanation) → Q151885 (concept) → Q5891 (philosophy) ← Q1799072 (method) ← Q46857 (scientific method)
Configuration	**Path found for Query B ⟨Q81938 (pain), Q482853 (vertebral column)⟩**
Optimized	Q81938 (pain) → Q408801 (celecoxib) → Q52849 (ankylosing spondylitis) → Q7577457 (spinal disease) → Q1979420 (human vertebral column) → Q482853 (vertebral column)
Semantics-Only	Q81938 (pain) → Q169872 (symptom) ← Q12136 (disease) ← Q1595418 (remedy) ← Q179661 (treatment) ← Q701216 (pharmacotherapy) ← Q12140 (medication) ← Q11190 (medicine) ← Q514 (anatomy) ← Q515083 (extremities) ← Q62513663 (lower limb) ← Q6027402 (human leg) ← Q23852 (human body) ← Q5170145 (core) ← Q160695 (torso) ← Q133279 (back) ← Q482853 (vertebral column)
Balanced	No path could be found within the entity limit.
Greedy	Q81938 (pain) → Q898407 (venlafaxine) → Q410142 (solute carrier family 6 member 4) → Q14330969 (brain development) → Q1073 (brain) → Q28947902 (cranium) → Q13147 (skull) → Q1377526 (axial skeleton) → Q482853 (vertebral column)
Uninformed	No path could be found within the entity limit.

algorithm's cost function, the introduction of a dual-entity query dataset for pathfinding in Wikidata, the optimization of the algorithm's three hyperparameters, and an evaluation of the algorithm on the said dataset with respect to the examined hyperparameter configurations. The provided re-implementation completes the picture.

Apart from the leads on future work mentioned above, one important point is the integration of the pathfinding algorithm in an end-to-end application where dual-entity queries can be entered, upon which the pathfinder is issued, such that a knowledge panel explaining the relationship between the query entities can be generated and finally be displayed. In this regard, it has to be investigated which kind of presentation yields the best user experience. For instance, an actual text describing the relationship could be generated using natural language generation techniques based on the found paths. At the same time, graph-based visualizations are conceivable, as well.

Instead of adopting the cost function introduced in [2], one could investigate alternative cost functions that also take the meaning of the predicates into account. Intuitively, predicates that express taxonomic relations might be more accessible for non-expert users than more specialized predicates.

Finally, a point raised in [2] should also be repeated here, namely how multi-entity queries that mention more than two entities could be served. While the obvious option is to simply issue a pathfinder between all pairs of entities and concatenate the resulting paths, an approach that tries to identify an entity that connects the query entities with minimal global concept drift is also conceivable, for example.

References

[1] Singhal A. Introducing the Knowledge Graph: Things, not Strings. Official Search Blog. 2012. https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html (accessed 2023/05/26).

[2] Martin L, Boockmann JH, Henrich A. Fast Pathfinding in Knowledge Graphs Using Word Embeddings. In: Schmid U, Klügl F, Wolter D, editors. KI 2020: Advances in Artificial Intelligence - 43rd German Conference on AI, Bamberg, Germany, September 21-25, 2020, Proceedings. vol. 12325 of Lecture Notes in Computer Science. Springer; 2020. p. 305-12. Available from: https://doi.org/10.1007/978-3-030-58285-2_27.

[3] Kasneci G, Elbassuoni S, Weikum G. MING: mining informative entity relationship subgraphs. In: Cheung DW, Song I, Chu WW, Hu X, Lin J, editors. Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009. ACM; 2009. p. 1653-6. Available from: https://doi.org/10.1145/1645953.1646196.

[4] Reinanda R, Meij E, de Rijke M. Knowledge Graphs: An Information Retrieval Perspective. Found Trends Inf Retr. 2020;14(4):289-444. Available from: https://doi.org/10.1561/1500000063.

[5] Vrandecic D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Commun ACM. 2014;57(10):78-85. Available from: https://doi.org/10.1145/2629489.

[6] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Trans Syst Sci Cybern. 1968;4(2):100-7. Available from: https://doi.org/10.1109/TSSC.1968.300136.

[7] Cyganiak R, Hyland-Wood D, Lanthaler M; W3C. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. 2014. https://www.w3.org/TR/rdf11-concepts (accessed 2023/05/26).

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

[9] Harris S, Seaborne A; W3C. SPARQL 1.1 Query Language. W3C Recommendation. 2013. https://www.w3.org/TR/sparql11-query (accessed 2023/05/26).

[10] Jaradeh MY, Oelen A, Farfar KE, Prinz M, D'Souza J, Kismihók G, et al. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, November 19-21, 2019. Marina Del Rey, CA, USA: ACM; 2019. p. 243-6. Available from: https://doi.org/10.1145/3360901.3364435.

[11] Wikidata; Wikidata. Statistics. Wikidata. 2023. https://www.wikidata.org/wiki/Special:Statistics (accessed 2023/05/26).

[12] Dietz L, Kotov A, Meij E. Utilizing Knowledge Graphs for Text-Centric Information Retrieval. In: Collins-Thompson K, Mei Q, Davison BD, Liu Y, Yilmaz E, editors. 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR 2018). ACM; 2018. p. 1387-90. Available from: https://doi.org/10.1145/3209978.3210187.

[13] Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A. Advances in Pre-Training Distributed Word Representations. In: Calzolari N, Choukri K, Cieri C, Declerck T, Goggi S, Hasida K, et al., editors. Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA); 2018. p. 52-5. Available from: http://www.lrec-conf.org/proceedings/lrec2018/summaries/721.html.

[14] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms, 3rd Edition. MIT Press; 2009. Available from: http://mitpress.mit.edu/books/introduction-algorithms.

[15] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein J, Doran C, Solorio T, editors. 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; 2019. p. 4171-86. Available from: https://doi.org/10.18653/v1/n19-1423.

[16] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. Association for Computational Linguistics; 2019. p. 3980-90. Available from: https://doi.org/10.18653/v1/D19-1410.

[17] Song K, Tan X, Qin T, Lu J, Liu T. MPNet: Masked and Permuted Pre-training for Language Understanding. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual; 2020. p. 16857-67. Available from: https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

[18] Feurer M, Hutter F. Hyperparameter Optimization. In: Hutter F, Kotthoff L, Vanschoren J, editors. Automated Machine Learning - Methods, Systems, Challenges. The Springer Series on Challenges in Machine Learning. Springer; 2019. p. 3-33. Available from: https://doi.org/10.1007/978-3-030-05318-5_1.

[19] Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proc IEEE. 2016;104(1):148-75. Available from: https://doi.org/10.1109/JPROC.2015.2494218.

[20] Stroeml C; Microsoft. Simple: Simple(x) Global Optimization. GitHub. 2018. https://github.com/chrisstroemel/Simple (accessed 2023/05/26).

[21] TREC; National Institute of Standards & Technology. Text REtrieval Conference (TREC) Home Page. TREC. 2018. https://trec.nist.gov/ (accessed 2023/05/26).

[22] Cao ND, Izacard G, Riedel S, Petroni F. Autoregressive Entity Retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net; 2021. Available from: https://openreview.net/forum?id=5k8F6UU39V.

[23] Snoek J, Larochelle H, Adams RP. Practical Bayesian Optimization of Machine Learning Algorithms. In: Bartlett PL, Pereira FCN, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States; 2012. p. 2960-8. Available from: https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230010

Polyvocal Knowledge Modelling for Ethnographic Heritage Object Provenance

Sarah Binta Alam SHOILEE [a,1], Victor DE BOER [a,2], and Jacco VAN OSSENBRUGGEN [a,3]

[a] *Vrije Universiteit Amsterdam, the Netherlands*

ORCiD ID: Sarah Binta Alam Shoilee https://orcid.org/0000-0001-9458-8105, Victor de Boer https://orcid.org/0000-0001-9079-039X, Jacco van Ossenbruggen https://orcid.org/0000-0002-7748-4715

Abstract. Purpose: Information about biographies of museum objects (*object provenance*) is often unavailable in machine-readable format. This limits findability and reusability of object provenance information for domain research. We address the challenges of defining a data model to represent ethnographic cultural heritage objects' provenance, which includes multiple interpretations (*polyvocality*) of, and theories for, the object biography, chains of custody and context of acquiring.
Methodology: To develop a data model for representing the provenance of ethnographic objects, we conducted (semi-)structured interviews with five provenance experts to elicit a set of requirements. Based on these requirements and a careful examination of six diverse examples of ethnographic object provenance reports, we established a set of modelling choices that utilise existing ontologies such as CIDOC-CRM (a domain standard) and PROV-DM, as well as RDF-named graphs..
Evaluation: Finally, we validate the model on provenance reports containing six seen and five unseen ethnographic cultural heritage object from three separate sources. The 11 reports are converted into RDF triples following the proposed data model. We also constructed SPARQL queries corresponding to nine competency questions elicited from domain experts in order to report on satisfiability.
Findings: The results show that the adapted combined model allows us to express the heterogeneity and polyvocality of the object provenance information, trace data provenance and link with other data sources for further enrichment.
Value: The proposed model from this paper allows publishing such knowledge in a machine-readable format, which will foster information contextualisation, findability and reusability.

Keywords. Cultural Heritage, Provenance information, Polyvocality, Domain modelling, Knowledge Representation.

1. Introduction

More and more cultural heritage organisations are producing and publishing their data as Knowledge Graphs [1]. One of the reasons for adopting these technologies in this domain is that the graph structure allows to express heterogeneity of information [2], while facil-

[1]Corresponding Author: Sarah Binta Alam Shoilee, s.b.a.shoilee@vu.nl
[2]Corresponding Author: Victor de Boer, v.de.boer@vu.nl
[3]Corresponding Author: Jacco van Ossenbruggen, jacco.van.ossenbruggen@vu.nl

itating interoperability. However, producing machine-readable knowledge graphs from existing structured data and unstructured sources is not a trivial operation [3]. At the same time, museum professionals and external researchers continuously acquire new information about collection objects and existing information is then cast in a new light. Especially in the context of post-colonial challenges to ”decolonise the database”, heritage institutions are seeking out how to incorporate previously underrepresented voices in their practice, collections and information systems (cf. [4]).

Cultural heritage institutions, especially those with ethnographic collections, continuously (re)contextualise objects by learning new facts about objects' biographies [5]. The traditional method is through dedicated research on individual objects, known as *provenance research*. Cultural heritage object provenance describes an object's history of ownership and evidence of the legal status of an object [6]. It can also be used to form an assessment of the authenticity of an artefact and identify any unlawfully appropriated works [7]. In many cases, the details of this extensive research are not available as structured data but only reported in a narrative textual document, which limits the findability, reusability and interoperability of such information. Typically, the person who conducts the research or the institute they are representing reflects on object metadata [8]. Once the research is done, the researcher might update a limited amount of metadata in the collection management system. In this process, potentially multiple views are reduced to a single perspective of truth. For example, the same object can be annotated with term “war loot” or “legally owned property”, depending on the researcher's interpretation of the documentation on the war. Such an interpretation can be affected by personal, cultural or political context and is likely to change over time. More importantly, when more than one annotation has equal merit to be valid, it is necessary to preserve both interpretations in the metadata. To allow future researchers and professionals to investigate these multiple perspectives, the institutions' information systems needs to be able to preserve, maintain and deliver the different views of objects and their provenance data [9].

The Semantic Web as an information architecture and Knowledge Graphs as the data model are promising technologies of such polyvocal knowledge representations [9]. Its dispersed and networked nature makes it ideally suited to handle diverse opinions, while at the same time preserving competing views with the sources of their origin. We consider this research on representing multiple perspectives as an example of such polyvocal knowledge representation. We investigate to which extent existing Semantic Web solutions, such as named graphs and existing ontologies for provenance and heritage, are suitable to represent multiple perspectives in data.

This paper's contribution lies in examining how Semantic Web technologies, (i.e., named-graphs, PROV-DM) and existing cultural heritage ontology (i.e., CIDOC-CRM) can be combined into a single model to meet the polyvocal provenance requirements specified by domain experts. We first identify these requirements for representing multi-perspective ethnographic object provenance information. We propose how such information can be modelled and apply this model to eleven ethnographic objects demonstrating the expression of complex chains of custody of the object biographies while preserving data provenance. The solution preserves the polyvocality of such information when multiple alternate theories are available. The resulting knowledge graph is validated against the Competency Questions constructed from the requirement analysis.

2. Related Work

High-quality metadata is necessary to increase the accessibility and reusability of digital content. The metadata of a museum object must include detail about the object before it enters a museum, as well as details that are generated while the object is in the museum [10]. When modelling cultural heritage data, it must be represented in a usable way for non-technical users, such as cultural heritage experts, to query, review and reuse it. There has been extensive research done on modelling of cultural heritage metadata [1,3,11,12,13]. However, none of these works investigated how to model rich object provenance information, which typically ends up as textual reports only.

Both object-centric and event-centric ontologies have been developed to represent cultural heritage metadata. Research, however, found that an event-centric approach provides advantages for representing provenance or other temporal data [1,14]. The event-centric model represents knowledge through associated events, such as acquisition or production. An ISO standard since 2006, CIDOC-CRM [15,16] is an event-centric ontology which is designed for the cultural heritage sector to facilitate the integration and interchange. CIDOC-CRM can be used to model multiple instances of semantic information regarding a given reality by adding multiple information layers. However, research [17] has shown that by itself this is not an feasible solution for representing multi-perspective data as these multiple layers are simply information accumulation without mentioning data provenance. The authors argue that the data must be organised so researchers can easily find previous information and use it for new reasoning.

Conversations around multiple perspectives are taking place in the cultural heritage domain [18,19]. Dijkshoorn et al. [1] present six requirements for cultural heritage ontologies, one of these supports capturing multiple sources with possibly conflicting views while describing the same artefact. In their research, it has been shown that the Europeana Data Model [20] allows multiple records for the same object by using proxies. Proxies in EDM can, however, only depict objects on a general level by connecting a proxy to the object resource and not to a specific statement about that resource. A similar approach is adopted by Ockeloen et al. [21], who propose a proxy solution for representing biographical descriptions from different perspectives and sources.

Another solution for multi-perspective representation can be found using named graphs. Bizer et al. [22] state that information providers have different world views; therefore, a named graph allows different information providers to make different claims regarding the same entity. The advantage of named graphs is that it allows grouping a collection of triples to make statements on the whole set and can quickly be adopted when CIDOC-CRM is implemented in RDF. Having IRIs on the named graphs introduces the possibility of attaching data provenance to the graph itself.

While the need for multi-perspective representations of cultural heritage data is identified, the practical application is still challenging. This research identifies a possible solution for representing multi-perspective interpretations of cultural heritage object provenance that is based on the domain standards discussed above.

3. Requirement Analysis

This section describes the requirement analysis for representing multi-perspective representations of cultural heritage provenance. A more detailed account of this analysis is

Table 1. Overview of museology expert interviewees

Respondent	Role	Expertise
R1	Postdoctoral researcher	Objects from East Africa
R2	Junior provenance researcher	Object combined with human remains
R3	Senior provenance researcher	Objects from Central and Southern Africa
R4	Postdoctoral researcher	Objects collected in Missionary context
R5	Senior provenance researcher	Objects from Asia

found in [23]. We here present the main approach and the resulting list of requirements and competency questions.

3.1. Approach

To collect data requirements for multi-perspective representations of cultural heritage provenance, we conduct a problem analysis through focused interviews with domain experts, which is concerned with developing an understanding of the nature of the problem. Focused interviews are a basic requirement engineering tool, to investigate current problems and concerns. After identifying the requirements for the data model, we utilised them to construct the model (Section 4). Additionally, we elicited nine Competency Questions from the interview which we use to validate the model.

For the focused interviews, we recruited five Museology professionals who are involved in the Pressing Matter project[4] in different capacities (see Table 1 for an overview of the interviewees. The index mentioned for participants in the table will be used in the rest of the paper to indicate corresponding respondent). Pressing Matter is a Dutch project which investigates artefacts collected during the colonial period to support societal reconciliation with the colonial past. The professionals were chosen based on their varied experience, background, and working methods. Although they work on different collections of ethnographic objects, they all have experience with the current museum information system and are responsible for updating object metadata with provenance information. These professionals can be considered the end-users of the data model developed in this research.

Each participant completed a one-hour individual semi-structured interview, with all interviews following the same interview guide. The interview guide [24] is aligned with the objective of this research[5]. The interview addressed the proper representation of cultural heritage provenance data, covering (1) provenance research processes and challenges, (2) documentation of research, (3) representation of provenance information, and (4) the utility of such information. A pilot interview was conducted before the actual study, and its insights were incorporated in the next interviews. All interviews, except the pilot, were conducted via web conferencing.

3.2. Findings

We report on the main findings in three parts. First, how provenance research is conducted and documented. Second, the identified challenges and problems with current

[4] https://pressingmatter.nl/

[5] The interview guide can be found at https://doi.org/10.5281/zenodo.7437713

representations are presented. Third, the respondents' opinions on multi-perspective representations of cultural heritage provenance.

Provenance Research: Interview results suggest that there is no standard goal for provenance research. However, all respondents agreed that it helps in gaining a better understanding of where collections and objects come from, leading to better-documented collections. Respondents had different approaches to conducting research, with varying reliance on sources such as archives, libraries, and web searches. All respondents, however, begin their research from the museum's collection management system (CMS)[6]. Important to note that, two mentioned that the system often contains missing information and observational bias. Respondents also shared that there are no guidelines on how to represent provenance information. Typically, when the information goes beyond the CMS, a separate report is written. However, there is no efficient way to trace or find such information within the system except for the unstructured text report.

Problem with current representations: Participants agreed that current information representation in the CMS is problematic due to faulty, incomplete or unreliable information. Lack of digitisation of archival material (R5) and decentralisation of available materials for information (R3) are identified as major challenges. Current representations of provenance in the CMS do not match the complexity of provenance research (R4), and important relations among people, places, objects and event cannot be represented in a machine-readable way (R3). Another problem identified by all the respondents is that the current management system does not contain any data provenance information, making it difficult to trace provenance of statements previously made about an object.

Opinions on multi-perspective representation cultural heritage data: Respondents agreed that keeping nuance in object provenance information is important, as changing times and perspectives lead to new ways of perceiving information. Museum database records can be influenced by the dominant perspective of their time, such as colonial representations and language use, which may not align with current views (R2, R4). Acknowledging how objects were seen before can tell us something about collections (R4). If multiple versions of provenance exist, all should be represented, as provenance is rarely fully proven (R5). On the other hands, some respondents (R1, R3) argue that it's not practical to preserve all information, and it depends on the research goals. Another respondent (R2) notes the importance of distinguishing between information deemed more correct now versus prior research. They also agreed that their research is just one interpretation of an object's history; it is impossible to say that their research is the final interpretation of the history of an object.

3.3. Identified Requirements and Competency Questions

The overall requirements reported by the domain experts for a representation are presented in the list below, divided into three types: overall representation, object provenance (information identified as important related to the chain of custody of an object) and data provenance (about the cultural heritage provenance statements, such as sources used during the research).

Overall representation Domain experts report that:

[6] In this case, TMS https://www.gallerysystems.com/solutions/collections-management/

- *Digital Representation:* Provenance research is easier to conduct if cultural heritage data is digitally available.
- *Event-centric representation:* It is easier to identify relations between actors, objects and places when they are represented in an event-centric way.
- *Machine-readable:* Machine-readable representations are easier to access compared to when the provenance research is presented in a written report only.

Object provenance

- *Object data:* Representation of object title, object number, object category, material, part of which collection and its origin.
- *Object creation:* When, where, and by whom the object was created.
- *Actors:* The network of actors involved in the object's collection.
- *Locations:* Object acquisition and creation places as well as travel route.
- *Events and time periods:* Historical events or time-period, may provide context, including unethical acquisition of colonial objects.
- *Multiple descriptions:* If multiple views on an acquisition exist they should be noted to keep nuance.
- *Comments:* An event may need detailed comments and notes in natural language.

Data provenance

- *Provenance statement source:* Users should be able to review the sources/author for provenance statements.
- *Source:* Users should be able to find the source materials of object provenance.
- *Traceability to previous research:* Each version of object provenance research should include data provenance information.

3.3.1. Competency Questions

Competency questions (CQs) are questions in natural language that outline the knowledge and specify the constraints for knowledge representation [25]. The concept of competency questions was explained during the interviews, and the respondents were requested to come up with specific questions based on their own requirements. The individual CQs were then aggregated in this study. All participants agreed that it is crucial to keep track of the *people* who were engaged in object acquisition (collectors, traders...) to identify networks of individuals involved in the acquisition of an object (**CQ1, CQ2, CQ3**). It is also important to convey information about *dates and events*; for example, the date or occation the object was obtained (**CQ4**). This enables identifying networks of connected objects through historical events, which may collectively project on an objects' acquisition (**CQ5**). *Geographic locations* are important to determine which items were bought or sold in particular regions or countries (**CQ6**). The respondents also identified *data provenance* as a crucial part in their competency questions. The participants unanimously agreed that each claim about the objects' provenance must be documented to track previous studies (**CQ7**, **CQ8**, and **CQ9**). They also mentioned the need to revisit earlier provenance versions to acquire a complete picture of all previous studies. The full list of aggregated competency questions from the text above is shown in the first two columns of Table 2. In Section 5.2, we describe how these are used for the purpose of validation.

Table 2. Competency questions (Section 3.3) and corresponding SPARQL queries (Section 5.2)

<table>
<tr><th>ID</th><th>Question</th><th>SPARQL query</th><th>Answers CQ?</th></tr>
<tr><td>CQ1</td><td>Which persons were involved in the provenance of this object?</td><td>SELECT * WHERE {
?o a crm:E24_Physical_Human-Made_Thing .
?o crm:P49_has_former_or_current_keeper ?p.
?p rdfs:label ?lab}</td><td>Yes, demonstrated query answers if the intent is to find out actors involved in object biography as a formal keeper or owner.</td></tr>
<tr><td>CQ2</td><td>Which objects are collected by person A?</td><td>SELECT * WHERE {
?p a crm:E39_Actor .
{?act crm:P29_custody_received_by ?p.
?act crm:P30_transferred_custody_of ?o} UNION
{?act crm:P23_transferred_title_from ?p.
?act crm:P24_transferred_title_of ?o}}</td><td>Yes, query retrieves all objects ?o if associated with actor ?p through any collection activity.</td></tr>
<tr><td>CQ3</td><td>Is there a relationship between person A and person B?</td><td>SELECT ?p1 ?p2 WHERE {
?p1 a crm:E39_Actor .
?p2 a crm:E39_Actor .
?act1 a crm:E7_Activity.
?act1 ?prop1 ?p1.
?act1 ?prop2 ?p2.
FILTER (?p1 != ?p2)}</td><td>Yes, query demonstrates retrieval of two persons, involved through a shared activity with the same object.</td></tr>
<tr><td>CQ4</td><td>Which objects were collected in this geographical location?</td><td>SELECT ?o ?p WHERE{
?o a crm:E24_Physical_Human-Made_Thing .
{?act crm:P30_transferred_custody_of ?o}
UNION
{?act crm:P24_transferred_title_of ?o}
?act crm:P9_consists_of ?sub .
?sub crm:P7_took_place_at ?p. }</td><td>Yes, query demonstrates how to retrieve object with location when location</td></tr>
<tr><td>CQ5</td><td>Which objects were collected during this event?</td><td>SELECT DISTINCT ?obj ?event WHERE {
?event a crm:E5_Event .
?event crm:P9_consists_of ?act .
?m_act crm:P9_consists_of ?act .
{?m_act a crm:E8_Acquisition .} UNION
{?m_act a crm:E10_Transfer_of_Custody .}
?m_act ?p ?obj .
?obj a crm:E24_Physical_Human-Made_Thing. }</td><td>Yes, given historical event ?e, this query returns all the objects whose collection activity is relevant to this event</td></tr>
<tr><td>CQ6</td><td>Which objects were collected in this geographical location during this time period?</td><td>SELECT ?o ?p ?b_time ?e_time WHERE{
?o a crm:E24_Physical_Human-Made_Thing .
?act crm:P9_consists_of ?sub .
?sub crm:P7_took_place_at ?p.
?sub crm:P4_has_time-span ?time .
?t crm:P82a_begin_of_the_begin ?b_time .
?t crm:P82b_end_of_the_end ?e_time.
{?act crm:P30_transferred_custody_of ?o}
UNION
{?act crm:P24_transferred_title_of ?o}}</td><td>Yes, the query returns all the objects with known geographic location and time-period</td></tr>
<tr><td>CQ7</td><td>Which source states this statement?</td><td>SELECT * WHERE{
Graph ?g {?s ?p ?o . }
?g (a|!a)+ ?g_o .
?g_o a prov:Entity .</td><td>Yes, the query returns all statements with their associated named graph and data provenance for the named graph</td></tr>
<tr><td>CQ8</td><td>Who or which institution conducted this research?</td><td>SELECT ?r ?a1 ?a2 WHERE {
?r a prov:Activity .
?r prov:wasAssociatedWith ?a1 .
OPTIONAL
{? prov:actedOnBehalfOf ?a2}}</td><td>Yes, given an research activity, the query returns agents or institution</td></tr>
<tr><td>CQ9</td><td>Which is the latest version of the provenance research?</td><td>SELECT ?act ?date WHERE
{ Graph ?g {?s ?p ?o .}
?g prov:wasDerivedFrom ?entity .
?entity prov:wasGeneratedBy ?act .
?act prov:endedAtTime ?date .
filter not exists {
?act prov:endedAtTime ?date1
filter (?date ?date1) }
} ORDER BY DESC(?date)</td><td>Yes, given a triple the query returns provenance report associated with it in descending publishing order.</td></tr>
</table>

4. Data Model

To further guide the modelling of object provenance information, we investigate this in the form of a case study concerning six ethnographic objects that are described in two different provenance reports "Provenance #1"[26] and "Provenance #2" [27], issued by the Dutch National Museum of World Cultures (NMVW)[7] to embed provenance of artworks into its practice and policy. They describe objects with rich provenance information elicited by extensive provenance research on these objects.

One object, RV-1148-1 is an elephant tusk that was part of the extensive provenance research on collections from Benin City. This provenance research aimed to assess the strength of connection to the military campaign led by British forces against Benin City in early February 1897; the research can be found in the report "Provenance #2" [27]. The rest of the objects (RV-2584-169a, RV-2334-1, RV-2334-2, RV-2334-3 and RV-2334-1) are from "Provenance #1" [26], where two different types of provenance information are found. On one side, RV-2584-169a is an interesting case-study because of having different possible theories of the acquisition and/or origin of the object, which left the researcher incapable of concluding on one theory with a high degree of certainty. On the other hand, the objects with id RV-2334-* are insightful because of their complicated chain of custody and links to different archival material. Despite the diversity in information, what is common in all the objects is that there have been discovered possible links to important historical events or time-periods, which projects unethical acquisition in the objects' chain of custody. The diversity in information, possible links to different historical events and time-periods, and available connections to different archival materials make these objects an ideal case study for the current research. These objects not only refer to the requirements identified by the previous section but also represent the complex nature of such knowledge. In the following subsection, we will describe our modelling choice for the proposed data model.

The first step is to reconstruct the information of these objects to identify essential statements related to the object's provenance from the textual report. This information is translated to provenance data that illustrates the key components considered necessary for representing the artefact's provenance. For this research, we investigated the use of CIDOC-CRM to ensure reusability.

4.1. Object

Our requirements indicate there are various types of knowledge about the object to represent. First-level object information is essential even if detailed provenance is unknown. As one respondent (R1) mentioned, the origin can be identified by knowing the materials used or the creation or collection date. CIDOC-CRM allows an object- and event-centric approach to co-exist by connecting instances directly to an object and also without an intermediary event. The domain experts call for representations of objects' inventory number, title, category or classification, material, collection and origin. Additionally, they request the possibility of attaching comments and descriptive notes.

Based on the participants' requirement of object description, this research reuses the specification mentioned for the function to express object collection information by

[7] http://wereldculturen.nl/

CIDOC-CRM guideline[8], with adjustments based on use-case requirements. The collection object itself is an instance of **E24 Physical Human-Made Thing**[9], with the following properties (and object classes).

- Inventory Id & Title: *P1 is identified by* (**E41 Linguistic Appellation** & **E42 Identifier**)
- Object classification: *P2 has type* (**E55 Type**)
- Textual Description: *P3 has note* (Literal)
- Related Person or Organization: *P49 has former or current keeper*, *P52 has current owner* (**E39 Actor**)
- Object Dimension: *P43 has dimension* (**E54 Dimension**)
- Material: *P45 consists of* (**E57 Material**)
- Represented Visual Concept: *P65 shows visual item* (**E36 Visual Item**)
- Image: *P138i has representation* (IRI)
- Collective Name/ Group: *P67i is referred to by* (**E33 Linguistic Object**)

Additional object information is best represented through events, i.e., **E7 Activity**, **E8 Acquisition**, **E12 Production** or **E10 Transfer of Custody**. Therefore, such information is connected with representative activities that are themselves connected to the artefact. For example, *P14 carried out by*, *P7 took place at*, and *P4 has time-span* are properties of the **E12 Production** event, where the object connects this activity with property *P108i was produced by*. On the other hand, properties such as *P28 custody surrendered by*, *P29 custody received by*, and *P30 transferred custody of* are not mentioned due to the complexity of such information and are modelled as part of provenance information.

4.2. Provenance Information

In addition to representing basic object information, the domain experts desired representations of detailed provenance information with known actors, locations and events. An object's provenance can be seen as a series of events where the custody of an object is transferred between different actors during time and places. The provenance of an object is mainly represented using two different entities in CIDOC-CRM. **E8 Acquisition** comprises the transfer of legal ownership from one or more instances of **E39 Actor** to another. In contrast, **E8 Acquisition** refers to legal ownership, thus the view that the change of owners is interpreted as a legal right, for example, object is purchased.

Common to all six objects considered here, there is at least one transfer of custody in their biography that is not seen as a legal right, namely when it was looted during the military campaign, purchased from illegal authority or receiving questionable gift. Therefore, using E8 Acquisition for modelling unethical ways of acquisition where the legal right is questioned may not be a appropriate. CIDOC-CRM separates legal ownership and physical custody. **E10 Transfer of custody** can be used to represent non-legal ways of acquisition, where a specific type of acquisition, such as theft, loot or gift, can be declared.

For the current modelling choice of selected objects, we used **E10 Transfer of Custody** to represent any illegal transfer of ownership and **E8 Acquisition** for legal cases. Any such activity (both E8 and E10) can further contain other activity(-ies) as the sub-activity(-ies) falls within the space-time volume of the main activity. This sub-activity is

[8]https://www.cidoc-crm.org/FunctionalUnits/object-collection-information
[9]In the following, we list ontology classes in **bold** typeface and properties in *italics*

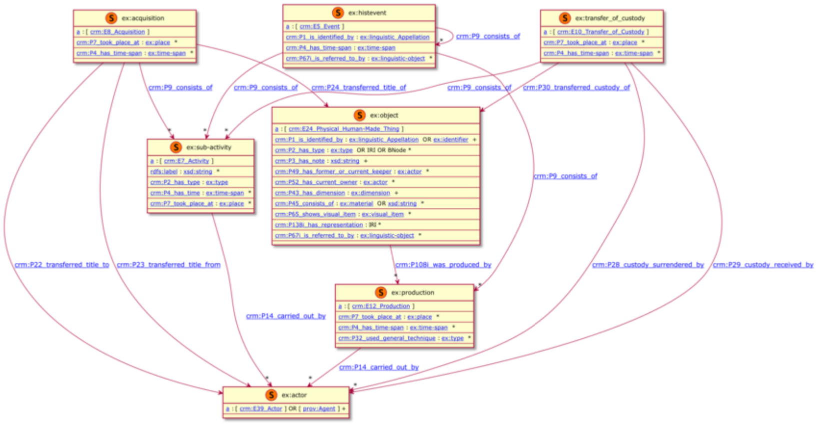


Figure 1. Ontology overview of cultural heritage object provenance modelling

an instance of **E7 Activity**, where the type of activity is further mentioned with *P2 has type* property and itself being connected with main activity with *P9 consists of* property. Consider, an object acquisition that occurred as a result of a government representative receiving a diplomatic gift and later transferring it to the museum. In that case, the acquisition of the collection consists of the activity of "receiving gift". This distinction between the main activity and "typed" sub-activity is made to achieve a level of abstraction across all objects, even when the transfer method of ownership is unknown.

Common to both E8 and E10 there is possibility to include a time-span, a location to the event and actors involved. Since both of them are sub-class of **E2 Temporal Entity**, time specification can be mentioned by *P4 has time-span* property. As subclasses of **E4 Period**, they can have *P7 took place* at properties with a **E53 Place** instance as object. For instances of **E8 Acquisition**, the properties *P23 transferred title from*, *P22 transferred title to* and *P24 transferred title of* are used to connect actors and objects to the activity. Fo instances of **E10 Transfer of Custody** the properties *P28 custody surrendered by*, *P29 custody received by* and *P30 transferred custody of* play that role.

Figure 1 visualizes the main entities of the ontology created, generated using the RDFShape[10] visualizer. This diagram only specifies the shape for the primary entities, less significant entities' detail is left out here for visual simplicity.

4.3. Data Provenance

Besides the *object provenance* information, the domain experts requested representation of *data provenance*. The representation of the data provenance is required for traceability of research. First, it concerns sources linked to each claim regarding the objects' provenance. Secondly, it relates to data provenance regarding the provenance research itself, including details on who did the study, for which institution, and when it was done.

Our solution for such representations is to use named graphs in combination with the PROV-DM [28] ontology. Named graphs can be used to attach provenance data and model context and scope assertions[11]. This provides the capacity to assess various as-

[10] https://rdfshape.weso.es/

[11] cf. https://www.w3.org/2009/07/NamedGraph.html

sertions made in a graph by the information providers [22]. In this case, a provenance researcher can identify a single source of knowledge containing various statements. It is up to information consumers to decide whether or not they can trust the information provider and how reliable the information is.

PROV-DM [28] is a conceptual model for modelling provenance, with PROV-O being a mapping to RDF[12] with proven applicability in the cultural heritage domain [21,29]. In our model it is used for attaching data provenance to a named graph IRI. A typical use case for PROV-DM is to achieve data quality, traceability and trustworthiness. The **Entity**, **Activity** and **Agent** classes are the building blocks for the model. According to the model, any physical, digital or conceptual things can a **prov:Entity**, where an **prov:Activity** is any action that occurs over a time-period, and **prov:Agent** is any actor who is responsible for the action. Therefore, in our case, the named graph containing all triples from the provenance research is of type **prov:Entity**, the provenance research activity itself is of type **prov:Activity**, and the institution or the person who are involved with this research is a **prov:Agent**. Representing provenance research and derived statements is one example of how we modelled different statements generated from different resources and activities. In Figure 2, the named graph ex:story#1 includes all data triples associated with object RV-2584-169a, which were generated during the research activity ex:provenance_research#2. This activity is an instance of **prov:Activity** in accordance with the PROV-DM ontology, enabling data provenance tracing for the generated triples.

4.4. Polyvocal Modelling

Data provenance modelling and named graphs can be used to group (CIDOC-CRM) triples that conform to a particular view of acquisition and distinguish them from other statements that conform to another view. Figure 2 shows how named graphs separate different (CIDOC-CRM) triples representing specific views of acquisition and how data provenance of such named graphs is specified using the PROV-DM ontology. When querying for a particular object provenance with SPARQL 1.1, triples stating object provenance can be returned without making any distinction in acquisition theory. By using the GRAPH keyword, information from only specified (provenance) graphs can be returned. This allows attaching a source, location and/or time period to that view and responsible agents/sources for a group of statements. Each named graph or triple collection is represented as a **prov:Entity**, typically derived from (*prov:wasDerivedFrom*) another **prov:Entity**(i.e., source) or generated by (*prov:wasGeneratedBy*) a **prov:Activity** (i.e., a domain research activity).

5. Results and Validation

This section reflects on the RDF triples generated from converting object provenance reports using the proposed model. Six objects were initially used for modeling decisions, and five more were randomly selected from the Pilotproject Provenance Research on Objects of the Colonial Era (PPROCE)[13] project to test the model's generalizability. The re-

[12] https://www.w3.org/TR/prov-o/

[13] https://www.niod.nl/en/projects/pilotproject-provenance-research-objects-colonial-era-pproce

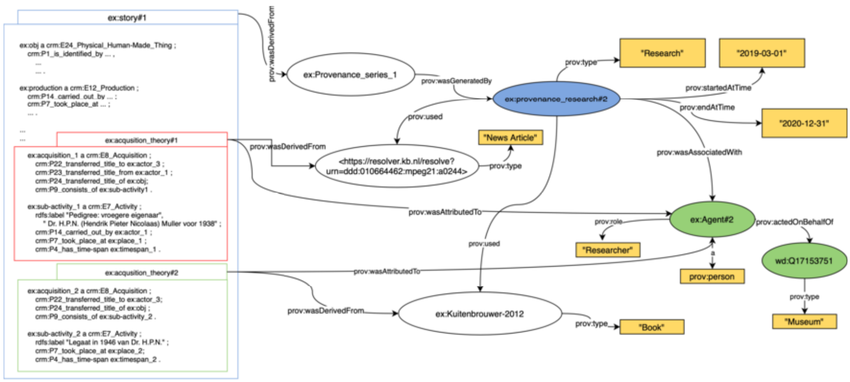


Figure 2. Multiple views or theories of Object acquisition are separated into named graphs and tagged with sources following the PROV-DM ontology. The transparent elliptic shapes represent **prov:Entity**, blue elliptical shapes represent **prov:Activity**, and green elliptical shapes represent **prov:Agent**.

sulting knowledge graph was validated against competency questions to ensure it adheres to domain requirements. All relevant files can be found in the Zenodo repository[14].

5.1. Statistics of Resulting knowledge graph

Following the modelling choices stated in the previous section and based on the information from the provenance reports, we first model the descriptions and provenance data of the six selected objects those were initially chosen to construct the model. For convenience, we are going to refer to these six objects as *construct object* in the rest of the paper. The resultant knowledge graph contains 1,786 triples spread across 31 named graphs. These named graphs contain either entire object metadata triples or triples generated by a single source of information. Additional named graphs were created to represent different views of the same entity with their source of information. The statistics of the resultant graph is given in the second column of Table 3. More statistics can be found in the Zenodo repository(/construct/entity_stat.csv) where the knowledge graphs itself is also available as TriG files in the data folder.

We also modeled the reports of 5 unseen objects from the PPROCE project , which we refer to as the "evaluation objects." The resulting knowledge graph contained 1,290 RDF triples spread across 27 named graphs, which is comparable to the construct objects set as we modeled only 5 objects' provenance data. The number of instances of different classes, such as **crm:E8 Acquisition**, **crm:10 Transfer of Custody**, **crm:E39 Actor**, **prov:Agent**, and **prov:Entity**, were consistent with the construct objects set. All TriG files and detailed statistics for this object set can be found in the "evaluation" folder of the Zotero repository.

The conceptual model is converted to ShEx[15] rules to maintain data consistency and shape. ShapeMap queries are used to validate each entity against the proposed ontology using the RDFshape[16] web tool. ShEx rules and ShapeMap queries can be found in the repository.

[14] https://doi.org/10.5281/zenodo.7437713
[15] ShEx, shape expressions, see https://shex.io/
[16] https://rdfshape.weso.es

Entity	Construct	Evaluation
Named graphs	31	27
crm:E24_Physical Human-Made Thing	6	5
crm:E12_Production	7	5
crm:E10_Transfer of Custody	7	7
crm:E8_Acquisition	12	8
crm:E7_Activity	21	12
crm:E5_Event	3	1
crm:E39_Actor	27	14
crm:E52_Time-Span	57	50
crm:E53_Place	11	10
prov:Activity	2	5
prov:Agent	20	18
prov:Entity	38	51
Total triples	1778	1290

Table 3. Number of triples for both the initial six construct objects and the five additional objects used for validation

5.2. Validation through Competency Questions

We validate the Knowledge Graphs using SPARQL queries to answer 9 competency questions provided by domain experts. The queries are listed in Table 2 (third column). Interpretation of the CQs and corresponding SPARQL queries are discussed below.

For **CQ1**, the listed query only matches if the intent is to find formal keepers or owners involved in the object's biography, but an alternate query is required to determine the exact capacity in which actors are involved in the object's provenance, as different activities are connected to objects with incoming and outgoing links. The alternate query is available in the supporting material.

CQ2 is interested in retrieving all objects that are connected with person A through a collection activity, i.e., **E8 Acquisition** or **E10 Transfer of Custody**. The query in Table 2 for **CQ2** retrieves both collection activities for an object and involved collectors; therefore answers the CQ accurately.

CQ3 can be answered in multiple ways. The listed query in the table retrieves two persons involved through a shared activity of the same object. If the intent is to find two actors linked with the same object, a separate query is needed (provided in supporting material). Nevertheless, computing all possible paths between two actors can be computationally expensive.

All activities can list its location using *crm:P7_took_place_at* properties. So, each collection activity is connected with the location if this information is known. The query given for **CQ4** targets to retrieve location when it is connected with sub-activity of Acquisition or Transfer of custody activity. Similar, queries can be written when the location is connected directly with activity. For detail, see Zenodo repository(validation_sparql.txt).

The query listed for **CQ4** retrieves location information when it is connected with a sub-activity of **E8 Acquisition** or **E10 Transfer of Custody** activity. Other queries can be written to retrieve location information when it is connected directly with the activity. More details can be found in the Zenodo repository under "validation_sparql.txt".

We answer **CQ5** by making multiple 'hops', since historical events are not directly connected with the objects, but rather can consist of activities concerning the object.

CQ6 is an extension of **CQ4** with a time-period specification. The same query used for **CQ4** can be reused for this one with temporal information. However, more advanced queries can be implemented to find temporal matches. A query for finding which object was collected from a given location withing a specific time-period is provided in the supplementary material.

All data statements are represented within one or more named-graph depending on source(s) and named graphs are connected to corresponding source(s) or responsible agent(s) acting on behalf of institution(s). The query for **CQ7** retrieves both sources that directly connects to data statements and sources that are connected through the associated activity; hence $(a|!a)$ expression is used. The alternative query to find out who/which institution makes this statement is given in the supplementary document.

The answer to **CQ8** is straightforward, as each research activity generating object provenance data is represented as an **prov:Activity**. Each **prov:Activity** is then connected to one or more **prov:Agent**, according to the PROV ontology, which can answer who/which institution is conducting the research.

The query for **CQ9** retrieves the associated named graph and **prov:Activity** responsible for any given statement. It lists all versions of these activities in descending order of execution time, and the latest version can be retrieved by specifying $LIMIT\ 1$.

In conclusion, we demonstrate that the proposed model can answer all nine questions, although some of them are too broad and require further interpretation to be answered through SPARQL queries.

6. Discussion

The query results and the implementation of real-world object provenance (both from seen and unseen report) confirm that the combination of CIDOC-CRM, PROV-DM and named graphs can be used to model the representation of object and data provenance. Technologically, we did not observe particular obstacles in representing ethnographic cultural heritage objects' provenance information in an interoperable manner. Nonetheless, it is essential to note that provenance research produces a mass of information; thus, unstructured data, i.e., written narrative report on a single object's biography may contain richer information. Additionally, there will always be a trade-off between expressivity and efficiency in digital humanities. Therefore, the representation of object provenance in a Knowledge Graph might not contain all the information recorded in the textual format regarding the provenance of an object. However, because the model supports representations of complicated networks between objects, people, places, and events utilizing the model, it projects an valuable overview to contextualize objects.

The provenance report summary can mostly be recorded using our model; however, we would like to highlight the interesting findings including the limitations and challenges encountered when modeling evaluation objects. Actor background and biography were not part of the ontology scope. The collection's context was preserved through historical events, dates, places of collection, and form of acquisition. Nevertheless, the textual narrative may provide additional information. The model does not distinguish between current custodian and current possessor and does not address predecessor relationships between organizations, such as mergers or renaming. These issues are beyond the scope of this paper and would require a deeper understanding of domain needs.

Polyvocality, observed through various theories of origin and acquisition, is crucial in determining the provenance of cultural heritage objects. Although the model supports polyvocal information representation, it does not prioritize one theory over another, especially when provenance reports may list and question information simultaneously. The model lacks the ability to assign weight to different statements, even if they contradict each other, due to the absence of an existential quantifier. Another important issue is that the current data model do not support information misrepresentation happened in the past. To preserve this information, the model places such statements under different named graphs and connects them with agents who made the statements. However, there is no means to indicate that this information is no longer considered valid. One solution is to use time-period information with named graphs to be able to refer to historical (event-based and provenance based) context.

To simplify the management of a cultural heritage object's chain of custody and to provide an abstraction over multiple objects, this study suggests using **E8 Acquisition** for legally recognized acquisitions and **E10 Transfer of Custody** for all other transfers. These entities can be further specified with sub-activities to define the type of ownership transfer, enabling institutions to model their specific notions of accession and deaccession. The International Council of Museums' documentation standard emphasizes the importance of using controlled terms to ensure consistent documentation [6], but the domain-standard vocabulary, AAT, lacks terms for unethical acquisitions.

7. Conclusion

Previous research identified a need for extending the domain ontology CIDOC-CRM to provide effective solutions for modelling multiple interpretations of cultural heritage object [17]. This study identifies requirements for modelling multiple perspectives on biographies of cultural heritage objects. After analyzing six distinct examples of ethnographic object provenance reports and considering the requirements, this paper proposes a data model that utilizes existing ontologies, i.e., CIDOC-CRM and PROV-DM, along with RDF-named graphs. Validation on six seen and five unseen objects confirms that the proposed model addresses complex chain-of-custody, data provenance, and multi-perspective representation requirements. We therefore conclude that the proposed data model allows to express cultural heritage object provenance in an interoperable manner.

In the field of heritage and humanities, and especially in the context of "decolonization" of the museums' databases, it is crucial that multiple (temporal, cultural and geographical) views from researchers, source communities and others, can be represented in the data structures. Although we focus in this research on ethnographic heritage collection's provenance information, the findings have implications on a more general provenance report to express such data polyvocality. Future work should incorporate information extraction tools to automate data conversion from textual reports of such knowledge that is inherently complex. The other possibility is facilitating the domain expert with easy tooling support to allow data modelling by themselves. Additionally, the provided model can be extended with methods to assign degrees of certainty to statements to allow data modellers to indicate the confidence levels of those statements.

Acknowledgements

The authors would like to thank the domain experts for their invaluable contributions. This work is partially based on previous work conducted in collaboration with Anne Coos van der Tang Eliasson [23]. This work was carried out in the context of the NWA funded project Pressing Matter (NWA.1292.19.419).

References

[1] Dijkshoorn C, Aroyo L, Van Ossenbruggen J, Schreiber G. Modeling cultural heritage data for online publication. Applied Ontology. 2018;13(4):255-71.

[2] Bizer C, Heath T, Berners-Lee T. Linked data: The story so far. In: Semantic services, interoperability and web applications: emerging concepts. IGI global; 2011. p. 205-27.

[3] Knoblock CA, Szekely P, Fink E, Degler D, Newbury D, Sanderson R, et al. Lessons learned in building linked data for the American art collaborative. In: International Semantic Web Conference. Springer; 2017. p. 263-79.

[4] Turner H. Cataloguing culture: Legacies of Colonialism in museum documentation. UBC Press; 2020.

[5] Modest W. Ethnographic Museums and the Double Bind. Matters of Belonging. 2019.

[6] Grant A, Nieuwenhuis J, Petersen T. International guidelines for museum object information: the CIDOC information categories. International Committee for Documentation of the International Council of . . . ; 1995.

[7] Campfens E. The Bangwa queen: artifact or heritage? International Journal of Cultural Property. 2019;26(1):75-110.

[8] Turner H. Organizing Knowledge in Museums: A Review of Concepts and Concerns. Knowledge Organization. 2017;44(7).

[9] van Erp M, de Boer V. A polyvocal and contextualised semantic web. In: European Semantic Web Conference. Springer; 2021. p. 506-12.

[10] Choy SC, Crofts N, Fisher R, Lek Choh N, Nickel S, Oury C, et al.. The UNESCO/PERSIST Guidelines for the selection of digital heritage for long-term preservation. UNESCO Digital Library; 2016. https://unesdoc.unesco.org/ark:/48223/pf0000244280.

[11] de Boer V, Wielemaker J, van Gent J, Oosterbroek M, Hildebrand M, Isaac A, et al. Amsterdam museum linked open data. Semantic Web. 2013;4(3):237-43.

[12] Hyvönen E. Cultural heritage linked data on the semantic web: Three case studies using the sampo model. VIII Encounter of documentation centres of contemporary art: open linked data and integral management of information in cultural centres Artium, Vitoria-Gasteiz, Spain, October. 2016:19-20.

[13] Mouromtsev D, Haase P, Cherny E, Pavlov D, Andreev A, Spiridonova A. Towards the Russian Linked Culture Cloud: Data Enrichment and Publishing. In: Gandon F, Sabou M, Sack H, d'Amato C, Cudré-Mauroux P, Zimmermann A, editors. The Semantic Web. Latest Advances and New Domains. Lecture Notes in Computer Science. Springer International Publishing; 2015. p. 637-51.

[14] Goerz G, Albers L. Representing place in space and time-methodological aspects in modelling the provenance of cultural heritage knowledge. In: Provenance of Knowledge. Proceedings CIDOC Conference; 2018. .

[15] Doerr M. The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. AI magazine. 2003;24(3):75-5.

[16] Bikakis A, Hyvönen E, Jean S, Markhoff B, Mosca A. Special issue on Semantic Web for Cultural Heritage. Semantic Web. 2021;12(2):163-7.

[17] Van Ruymbeke M, Hallot P, Billen R. Enhancing CIDOC-CRM and compatible models with the concept of multiple interpretation. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2017;2.

[18] Modest W, Lelijveld R. Words Matter: An Unfinished Guide to Word Choices in the Cultural Sector. National Museum of World Cultures; 2018. https://www.materialculture.nl/en/publications/words-matter.

[19] Captain E, Osbourne A, Somers Miles R, Tzialli E. Inward Outward, Critical archival engagements with sounds and films of coloniality. 2020 Inward Outward Symposium. 2020.

[20] Isaac A, et al.. Europeana data model primer. Europeana; 2013.

[21] Ockeloen N, Fokkens A, Ter Braake S, Vossen P, De Boer V, Schreiber G, et al. BiographyNet: Managing Provenance at Multiple Levels and from Different Perspectives. In: LISC@ ISWC. Citeseer; 2013. p. 59-71.

[22] Carroll JJ, Bizer C, Hayes P, Stickler P. Named graphs, provenance and trust. In: Proceedings of the 14th international conference on World Wide Web; 2005. p. 613-22.

[23] van der Tang Eliasson AC. Representation of various perspectives for cultural heritage provenance. Master thesis. Vrije Universiteit Amsterdam; 2022.

[24] Castillo-Montoya M. Preparing for interview research: The interview protocol refinement framework. The qualitative report. 2016;21(5):811-31.

[25] Wiśniewski D, Potoniec J, Ławrynowicz A, Keet CM. Analysis of ontology competency questions and their formalizations in SPARQL-OWL. Journal of Web Semantics. 2019;59:100534.
[26] Johnson S, Veys FW, editors. *(Foreword by Henrietta Lidchi)* Provenance. vol. 1. Nationaal Museum van Wereldculturen; 2020.
[27] Veys FW, editor. *(Foreword by Henrietta Lidchi)* Provenance. vol. 2. Nationaal Museum van Wereldculturen; 2021.
[28] Moreau L, Missier P, Belhajjame K, B'Far R, Cheney J, Coppens S, et al. PROV-DM: the PROV data model technical reports. World Wide Web Consortium. 2012.
[29] Sandusky RJ. Computational provenance: Dataone and implications for cultural heritage institutions. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE; 2016. p. 3266-71.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230011

Virtual Reality Based Access to Knowledge Graphs for History Research

Julia BECKER [a] Mario BOTSCH [b] Philipp CIMIANO [a,1] Melanie DERKSEN [b]
Mohammad Fazleh ELAHI [a] Angelika MAIER [a] Marius MAILE [a]
Ingo Oliver PÄTZOLD [a] Jonas PENNINGROTH [a] Bettina REGLIN [a]
Markus ROTHGÄNGER [a] Silke SCHWANDT [a]

[a] *Bielefeld University*
[b] *TU Dortmund University*

Abstract. Purpose: Knowledge graphs have so far been intensively used in the cultural heritage domain. Current interaction paradigms and interfaces however are often limited to textual representations or 2D visualizations, not taking into account the 4D nature of data. In digital history in particular, where events as well as geographical and temporal relationships play an important role, exploration paradigms that take into account the 4D nature of event-related data are important, as they have the potential to support historians in generating new knowledge and discovering new relationships. In this paper, we explore the potential of virtual reality as a paradigm allowing digital humanities researchers, historians in particular, to explore a semantic 4D space defined by knowledge graphs from an egocentric perspective. **Methodology:** We present *eTaRDiS*: a virtual reality based tool supporting immersive exploration of knowledge graphs. We evaluate the tool in the context of a task in which historians and laypersons with a history background explore DBpedia and Wikidata. We report results of a study involving 13 subjects that interacted with the data in eTaRDiS in the context of a specific task, in order to gain insights regarding the interaction patterns of users with our system. The usability of the tool was evaluated using a questionnaire including questions from the System Usability Scale (SUS) in addition to task-specific questions. **Findings:** The usability evaluation showed that our tool achieved an overall SUS score of 71.92, corresponding to a ‘satisfactory’ rating. While the mean score reached with laypersons with a history background was quite high with 76.0, corresponding to a rating of ‘excellent’, the score for historians was lower with 69.4, corresponding to a ‘sufficient to satisfactory’ rating. A qualitative analysis of the interaction data revealed that participants quickly identified the relevant information in the tasks using a variety of strategies and taking advantage of the features provided in eTaRDiS. **Value:** eTaRDiS is to our knowledge the first virtual reality based exploration tool supporting the exploration of knowledge graphs. The findings of the usability evaluation and the qualitative analysis of exploration patterns show that the system could potentially be a valuable tool for allowing digital humanities researchers to explore knowledge graphs as a way to discover new relationships between historical events and persons of interest.

Keywords. knowledge graphs, VR, digital humanities, digital history, linked data, semantic web

[1] Corresponding Author: Philipp Cimiano, cimiano@cit-ec.uni-bielefeld.de

1. Introduction

Knowledge graphs have been shown to be useful in the cultural heritage (CH) domain to connect and access large collection databases originating from multiple sources [1]. Hyvonen et al. [2], for example, have developed the SAMPO data model which provided the basis for implementing multiple data portals to access information about cultural heritage. With this model, content can be made semantically interoperable to support access, search, and discovery, thus overcoming boundaries of data silos. While interoperability is an important dimension in providing access to CH data, investigating by which access paradigms digital humanities researchers can be supported in answering their genuine research questions using knowledge graphs is also an important research avenue.

In this paper, we are in particular concerned with how to leverage knowledge graphs for the work of historians. Conceptions of history are widely shared as narratives that decisively depend on socio-cultural frames of reference and systems of representation [3]. Exploring and understanding these systems is a central task of the humanities and cultural studies, with the goal to communicate a spatio-temporal relationship, e.g. a historical moment, which equals the construction of meaning and ultimately knowledge. In history, agents, places, and events and their relationships play an important role. Events in particular can be regarded as 4D objects as they take place in a particular geographical location and have a temporal extension. Agents in turn can be located at different places at different times and places can undergo changes over time. It is thus key that tools that provide access to knowledge graphs to historians account for the 4D nature of relevant entities, allowing to explore the spatio-temporal relationships between entities.

The idea of using virtual reality technology (VR) in the CH domain is not fundamentally new. VR has been applied to the visualization of historical artifacts or full museums to enable a more interactive exploration of such artifacts or museum collections [4,5,6]. Our approach differs from the *virtual museum* paradigm in that we do not follow the paradigm of 'bringing history to life' but aim to facilitate the exploration of historical data networks. Jósza et al. [7] create 4D spatio-temporal models of large dynamic urban scenes containing various moving and static objects. Although their work has not been carried out in the CH domain, it shows that VR is an adequate medium to visualize and explore spatio-temporal 4D data.

Knowledge graphs built from Linked Open Data (LOD) such as DBpedia and Wikidata can provide a basis for the construction of different views on networks. From these knowledge graphs, the resources of historical events, persons, places, etc., and related multimodal data can be retrieved and organized in a network for analysis in historical studies. Using the three-dimensional space in immersive scenarios makes it possible to explore new spatio-temporal relations of historical events, persons, etc. in virtual space. In this way, it becomes possible to observe the actions of people in a network of space and time that makes new connections visible [8, p. 8]. According to Ayers, the visualization of time happens through the visualization of the movement of events and people in space [8]. This perception is central to the construction of events [9, p. 193] and persons [10], which can also be interpreted as specific spatio-temporal relations.

Towards the goal to leverage knowledge graphs for the work of historians, we present a new approach to the exploration of knowledge graphs that relies on VR to create an immersive space in which the spatio-temporal connections between entities can be explored. Our approach allows to explore the semantic neighborhood of entities from different perspectives: focusing on entities that are geographically close, temporally close, or semantically close, depending on the research question.

Specifically, we present eTaRDiS, a VR application to enable novel spatio-temporal access to knowledge graphs that allows exploring the neighborhood of nodes by filtering according to different aspects: time, location, and semantic distance in particular. As a VR tool, eTaRDiS offers a new perspective on spatio-temporal relationships by deconstructing existing historical narratives, allowing the user to explore, compare, and (re)order abstract knowledge fragments, i.e. data from a knowledge graph. As a consequence, eTaRDiS can be framed as a "possibility space" [11], a sort of laboratory for data and an immersive exploration environment. An initial expert review of our approach gave us valuable insights into the use cases and requirements that guided our developments [12].

The application consists of two main views: the *Hub* (see section 3.2.1) and the *Fragmentarium* (see section 3.2.2). The Hub allows a user to select a dataset to explore and a node to start the exploration. The Fragmentarium allows to explore the neighborhood of a given node; we call such neighborhoods that are defined according to some distance measure *historical fragments*. The neighborhoods are visualized in VR as labeled spheres floating around the user. Selecting one of the related data points, in turn, updates the visualization by showing those data points related to the newly selected one. This enables the exploration of the network of related agents, places, events, etc. according to one's own individual interest.

As a main contribution of this paper, we describe the design and implementation of the eTaRDiS system. We further present the results of a user study involving 13 participants. As main result, the study showed that users generally found it easy to identify the relation between historical fragments using the features of the eTaRDiS VR application and by exploring the semantic neighborhood of fragments. A qualitative analysis of the interaction revealed that participants used different strategies to solve the tasks, tested the boundaries of the rules in our virtual environment, and liked the playful approach to exploring historical data. In the study, our tool achieved an overall System Usability Scale (SUS)[13] score of 71.92, corresponding to a 'satisfactory' rating. Further analysis showed that the self-assessment of users regarding their usage of digital media had an influence on the SUS score. Participants who rated themselves as 'good' to 'extremely good' at using digital media have a higher mean SUS score of 77.8 (a rating of 'satisfactory to excellent') compared to the participants who rated themselves as 'very bad' to 'not bad' at using digital media (mean SUS score: 58.8).

2. Related Work

Virtual reality (VR) has been shown to be a valuable tool for interaction with networks in several areas, especially in the natural sciences. For example, Pirch et al. [14] developed the 'VRNetzer platform' for the interaction with large network structures of genes to identify genes indicating rare diseases (see also the work of Buphamalai et al. [15]). Some applications, namely HisVA [16], VaiRoma [17], and POLIS [18], have been implemented for the exploration and analysis of data from the Semantic Web through 2D visualizations. However, no VR applications have been developed in the CH domain and the digital humanities to explore existing knowledge graphs.

According to Kidd [19], there is an immersive turn in museums and heritage contexts, one consequence of which has been a broader range of digital and other media utilized in museums. The goal of *immersive VR* is to give the impression to users that they are truly in a synthetic world that can be reached via using a Head-Mounted Display (HMD) because it allows users to focus on the display of projected VR without distraction [20]. For example, the immersive exploration of CH data in virtual and augmented reality has been proven useful in higher education in Art History [21]. Furthermore, Razum et al. [22] developed the virtual research environment TOPORAZ in which a 3D model of the main market of the city of Nuremberg is linked to a database in four time layers [22]. Although we also use immersive VR in eTaRDiS and make use of an HMD, our approach differs from the approach of Casu et al. [21] and Razum et al. [22] in that they follow the paradigm of ‘bringing history to life’, while in our approach VR supports the exploration of a network of historical fragments.

McIntire and Liggett discussed a bunch of data and information visualization applications to find out which kind of tasks might benefit or even suffer from a third dimension [23]. Among others, 3D proves advantageous to tasks including precise spatial localization of objects, complex imagery analysis, and manually interacting with data or virtual information. Furthermore, they found that a 3D visualization can provide performance improvements that correlated with cognitive benefits and that facilitate a better understanding of spatial and/or multidimensional data. Overall, they concluded that 3D is especially beneficial for data interpretation tasks.

Wagner Filho et al. [24] stated that exploring 3D scatter plots with an HMD leads to a smaller effort in finding information and offers a much larger subjective perception of accuracy and engagement as opposed to desktop applications, but may suffer from occlusions. Kraus et al. [25] investigated the impact of immersion on cluster identification tasks in scatterplot visualizations. Their results indicate that task performance differs between the investigated visualization design spaces in terms of accuracy, efficiency, memorability, sense of orientation, and user preference. In particular, the 2D visualization on the screen performed worse compared to the 3D visualizations with regard to the measured variables. The study shows that an increased level of immersion can be a substantial benefit in the context of 3D data and cluster detection. Overall, they state that virtual environments can indeed provide suitable design spaces for abstract visualizations such as scatterplots. Furthermore, it became apparent that getting an overview of three-dimensional data can be enhanced by means of VR due to a more natural navigation, and better orientation and memorability capabilities.

Wagner et al. [26] investigated the effect of exploration mode and frame of reference in immersive analytics. They found that egocentric exploration of space significantly reduced mental workload. Exocentric exploration, in turn, improved performance on some tasks. They concluded from their research that generally a room-scaled environment should be favored if the necessary space is available; both the egocentric and exocentric perspectives should be offered so that users can decide whether and when to switch between these perspectives based on their interests.

It can be concluded from previous work that VR seems to be a suited tool to provide access to historical data from multiple dimensions and perspectives, leading to a better understanding of the data and of the corresponding relationships inherent in it. The above-mentioned studies have substantially influenced the design of our application, in particular the choice of providing an egocentric perspective to the user at initiation in order to avoid occlusions. According to Wagner et al. [26], we let the user transform the initially room-scaled visualization to enable the switch to an exocentric perspective.

3. Design and Implementation of eTaRDiS

This section describes the design and implementation of eTaRDiS. First, we provide an overview of the types of users that the system is intended to support and how their needs differ. We then describe the actual implementation of the system, emphasizing our key design choices to meet the needs of our target users.

3.1. User Groups

Opening up the virtual space for the field of humanities and cultural studies also meant designing eTaRDiS to be intuitively usable and making it easily accessible to historians, laypersons interested in history, and people with an affinity for technology and an interest in VR. The eTaRDiS is intended for three stereotype users: Historians that are experts in a specific area of historical scholarship or that are academically active in that area, laypersons who are interested in history and cultural studies, and users who have no affinity for dealing with cultural or historical data. In our user study, we investigate the perspective of historians and laypersons.

The user group of historians knows research discourses and their contexts. In the composition of the sub-dataset, this group is able to translate their research perspectives in a very fine-granular way into the filter options offered by eTaRDiS and thus operationalize their questions. Exploration in the Fragmentarium not only provides an overview but also encourages them to change their perspective on a topic. The user group of laypersons includes not only people who have a basic affinity for history but also those who are on their way to a degree in the field of cultural and historical studies. In their use of eTaRDiS and their choice of filters, they are clearly more open and less specific than described in the first user scenario because they go into eTaRDiS with a broader view. It can be assumed that they use the insular knowledge they have acquired over the years and select a topic focus for their exploration against this background. They are concerned with the controlled reduction and expansion of complex units of information. They use their time in the Fragmentarium to gain overviews and consume the (text and image) data deposited for the information much more intensively.

3.2. Implementation

The eTaRDiS system consists of a VR application and a backend database. The VR application has been implemented using the *Unity* game engine[2], so that it supports the visualization of and interaction with knowledge graphs in a 3D immersive environment. The VR application consists of two virtual spaces: the *Hub* and the *Fragmentarium*. These spaces are clearly separated by different purposes and forms of interaction.

[2]Although we specialized our developments on the *HTC Vive*[3], the underlying concepts are applicable to other VR hardware as well.

[3]`https://www.vive.com/de/product/vive-pro/`

3.2.1. Hub

The Hub serves as a central retreat within the virtual environment of eTaRDiS in which basic settings such as language, type of keyboard input, etc. can be edited. In addition, the Hub is the starting point for virtual data exploration. An appropriate interface enables the user to filter the data to create a sub-dataset adequately tailored for individual (research) interests (see Figure 1), which can then be subsequently visualized and explored in the subsequent room, the Fragmentarium.

In order to allow users to get an overview of selected data, information about the filtered sub-dataset is dynamically displayed on a dashboard to allow a user to filter the dataset according to individual interests. In addition to a display of the total amount of nodes (*fragments*) in the created sub-dataset, temporal and spatial information is shown. Statistics are displayed for the entire sub-dataset, and the spatial distribution of the individual fragments becomes visible on a world map. Based on this information, initial considerations can already be made for selected filters.

In addition to filtering, the Hub interface allows loading *saved points* and *presets*. Saved points allow a user to continue a session at a later point in time and thus to start at the same point in the Fragmentarium, while presets can be used to save and load defined filters. presets can also be edited later.

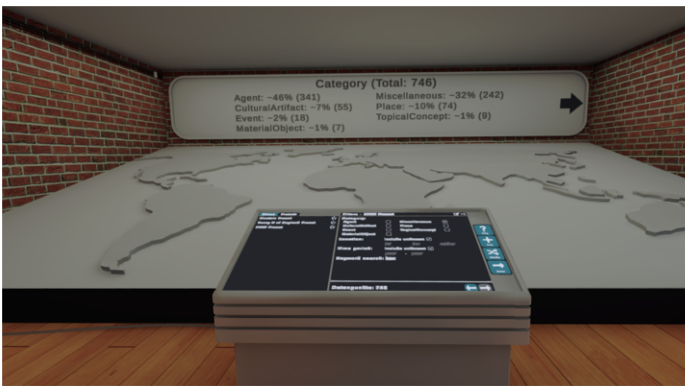


Figure 1. Filtering the dataset dynamically updates the statistical information on a dashboard in the Hub.

3.2.2. Fragmentarium

The Fragmentarium is the interface allowing a user to explore the neighborhood of a node in the knowledge graph selected in the Hub. The visualization consists of a historical main fragment and related fragments. The user is standing on the main fragment, represented as a colored circular platform, and is therefore intended to explore the scene from an immersive egocentric viewpoint. Related fragments are represented as labeled colored spheres that surround the user (see Figure 2). While the color indicates the fragment's category (e.g., agent, event), the sphere's size is related to the number of sources (i.e., `dbo:wikiPageExternalLinks`) available in the corresponding DBpedia article.

For the arrangement of the related fragments, two kinds of distance measures are used. The first one is used to position the spheres on a spherical shell around the user so that fragments with a low dissimilarity are close to each other. The second one determines their individual distances to the center according to their dissimilarity to the main fragment. The minimal and maximal distances to the center and thus to the user are 1.7 m and 3 m. A high dissimilarity results maximally in a distance of 3 m to the center.

To position the spheres on a spherical shell, we rely on the *UMAP* algorithm [27], which is a dimension reduction technique. With the dissimilarity calculation of the related fragments among themselves as its input metric and the haversine as its output metric, UMAP finds a suitable arrangement of the related fragments around the user. To prevent dislocations of the neck, in our case, the UMAP algorithm gets two additional artificial points in its optimization process that are fixed on the poles and constantly repel all the other data points, which bypasses constellations in which spheres accumulate directly above or beneath the user.

We rely on four attributes for which we calculate dissimilarities: i) temporal (t), ii) spatial (s), iii) semantic (sem), and iv) categorical (c). Users can define the neighbourhoods to explore by arranging other nodes that are closest with respect to these different dissimilarities. We first compute pairwise distances for the related fragments in one visualization for each of the above-mentioned attributes and afterwards, scale them per attribute a to an interval from 0 (which means *similar*) to 1 (which means maximal *dissimilar*) to get a normalized measure of dissimilarities for each individual attribute. If there is no data available for a fragment $n \in F$ and one of its attribute values a_n, then the value is considered invalid. More formally, for a visualization with the related fragments F, the dissimilarity *diss* for two fragments $i, j \in F$ regarding one attribute $a \in \{t, s, sem, c\}$, with a_i, a_j being their corresponding attribute values, is defined as:

$$\mathrm{diss}_a(i,j) = \begin{cases} \frac{\mathrm{dist}_a(i,j) - \min_{k,l} \mathrm{dist}_a(k,l)}{\max_{m,o} \mathrm{dist}_a(m,o) - \min_{k,l} \mathrm{dist}_a(k,l)}, \text{ with } k,l,m,o \in F, k \neq l & \text{if } a_i, a_j \text{ are valid,} \\ 0 & \text{if } a_i, a_j \text{ are invalid,} \\ 1 & \text{otherwise.} \end{cases}$$

Note that two related fragments may have different dissimilarity values after the scaling process. This is due to the fact that two fragments may appear hand in hand for various main fragments which might have distinct sets of related fragments with different minimal and maximal distances, potentially leading to different scalings. We describe the way we compute the different distances in what follows:

Temporal distance: For the temporal distance dist_t between two fragments, the minimal separating interval regarding their time periods t_i and t_j is considered. If there are time periods provided that do not overlap, then the difference in days is taken into account. If their periods overlap, or if no time period is given for both fragments, their distance is 0.

Formally, a fragment's time period t_i is an interval $\left[t_{i,start}; t_{i,end}\right]$ starting with a start date $t_{i,start}$ and ending with end date $t_{i,end}$, each of which is represented in days (the number of days that have elapsed from common era to the corresponding date associated with the fragment). Two time periods t_i and t_j overlap, if their intersection is not empty: $t_i \bigcap t_j \neq \emptyset$. Their temporal distance in days $\mathrm{dist}_t(i,j)$ can be calculated as follows:

$$\mathrm{dist}_t(i,j) = \begin{cases} \min\left\{||t_{i,start} - t_{j,end}||, ||t_{j,start} - t_{i,end}||\right\} & \text{if } t_i \bigcap t_j = \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Spatial distance: The spatial distance dist_s between two fragments is defined as the shortest distances of the corresponding locations. Since locations are represented in pairs of longitudes and latitudes (ϕ, θ), the distance $dist_s$ between two locations (ϕ_i, θ_i) and (ϕ_j, θ_j) can be calculated via the great circle distance with $r_{earth} \approx 6371,009$ km being the earth's radius:

$$\text{dist}_s(i,j) = r_{earth}\ \arccos\left(\sin(\theta_i)\sin(\theta_j) + \cos(\theta_i)\cos(\theta_j)\cos(\phi_i - \phi_j)\right).$$

Semantic distance: We make use of the Sentence Mover's Distance by Clark et al. [28] which is an evaluation method for multi-sentence texts. As sentence vectors, we use a fine-tuned sBERT model [29] based on the pre-trained model 'all-MiniLM-L12-v2'[4]. We applied the Sentence Mover's Distance to the abstracts as textual descriptions of entities, and use uniform weights for the sentences in the calculation of the distance.

Categorical distance: The categorical distance is per definition a dissimilarity and estimates to 1 if two fragments do not share the same category. Otherwise, it is set to 0. Given the categories of two fragments, c_i and c_j, the categorical distance dist_c is computed as:

$$\text{dist}_c(i,j) = \begin{cases} 1 & \text{if } c_i \neq c_j, \\ 0 & \text{otherwise.} \end{cases}$$

The user can choose according to which dissimilarity the data should be arranged. In our scenario the distance measure that is used as the input metric for the UMAP algorithm takes the dissimilarity values of the categories of the related fragments among themselves into account while the second one makes use of the semantic dissimilarities between the related fragments and the main fragment. Thus, related fragments that share the same category appear in clusters around the user and related fragments that have a high semantic dissimilarity to the main fragment are spatially distant to the center. Therefore, the visualization gives clues regarding the relationship between the main fragment and its corresponding related fragments and regarding the interrelationships between related fragments at one glance.

3.3. Design Principles

The features in the Hub and the Fragmentarium were developed to meet the requirements for the scholarly primitives by Unsworth [30] `comparing`, `annotating`, `discovering`, and `sampling`. These primitives, among others, are common methods of humanities researchers, basic functions common to scholarly activity across disciplines, over time, and independent of theoretical orientation [30][5].

The primitive of `comparing` is addressed by the so called *relation details* in the Fragmentarium, which show pairwise connectivity information for the nodes regarding space and time (see Figure 2). The `annotation primitive` is realized by allowing users to mark nodes to find them later again more quickly. If nodes are marked, their mark appears in the color of their respective category on the compass, which is a guidance line in the upper field of view (see Figure 3). Furthermore, for each node, a *detail window* can

[4] https://www.sbert.net/docs/pretrained_models.html

[5] Pacheco [31] gives an elaborate analysis of the scholarly primitives in the digital humanities, also from other researches over the years.

be opened with collapsable additional information like its *characteristics* (e.g., locations, time period, etc.), an image, and an abstract (see Figure 3). In that window, a space is reserved for annotation by allowing users to enter notes. The primitive `discovering` is addressed with the visualization and features in the Hub and the Fragmentarium as a whole. The user can apply filters or show statistics on datasets in the Hub. At the same time, this addresses the requirements of `sampling` because the filters in the Hub can be used to select data according to the user's interests. Selecting a new node of interest with a ray pointer attached to the user's controller in the Fragmentarium makes it the new *main fragment* and updates the visualization. Thus, the user is able to move from node to node while inspecting their properties and relations. A history shows the order of the main fragments visited so far (see Figure 3). Following the recommendations of Wagner et al. [26], we offer both the egocentric and exocentric perspective to the user, who is allowed to grab and transform the whole constellation of the related fragments (which includes translating, uniform scaling, and rotating). Thereby, the user may switch from the initial egocentric to an exocentric perspective. Finally, the application offers a help menu that explains all functionalities.

3.3.1. Backend

In order to implement the data backend, we rely on Neo4j[6] as a graph database designed to manage and query large knowledge graphs. We rely on DBpedia as the main data source for several reasons. First, DBpedia, being built on top of Wikipedia, is the most widely [32] used online encyclopedia, and one of the most prominent examples of truly collaboratively created content. The Wikimedia Commons[7] is part of the Wikipedia encyclopedia family that contains over twenty-five million audio, video, and image files [33], including scanned books, historically significant photographs, illustrative figures, and maps. Accessing historical data via several media plays an important role in their exploration and interpretation. Third, DBpedia is interlinked to other related information and also connected to other datasets [34] that allow us to explore historical data through semantic relatedness. In addition to DBpedia, we use Wikidata as it represents information on the time and location of a resource by a set of well-defined properties, which is crucial for historical data exploration.

First, we extract resources from DBpedia including Wikipedia text, images, video, and audio files. For each resource, we extract its connections in the forward and backward directions. After that, we find the equivalent entry in Wikidata, provided via the property `owl:sameAs` in a DBpedia resource, and retrieve temporal and spatial information.

In order to integrate data from DBpedia and Wikidata into a knowledge graph database in the context of historical studies, we have worked with historians in the eTaRDiS project to develop an appropriate database schema. We re-structured the classes of DBpedia into the seven categories: `Agent`, `Place`, `Event`, `Topical Concept`, `Material Object`, `Cultural Artifact` and `Miscellaneous`. The reason is that DBpedia classes were not designed for the context of historical data. For example, the Motherland Calls (i.e., a resource in DBpedia `res:The_Motherland_Calls`) is a statue representing *'Heroes of the Battle of Stalingrad'* but the class of the resource is `dbo:ArtWork` in DBpedia, which is re-structured as `Cultural Artifact` in our

[6] https://Neo4j.com/
[7] https://commons.wikimedia.org/wiki/Main_Page

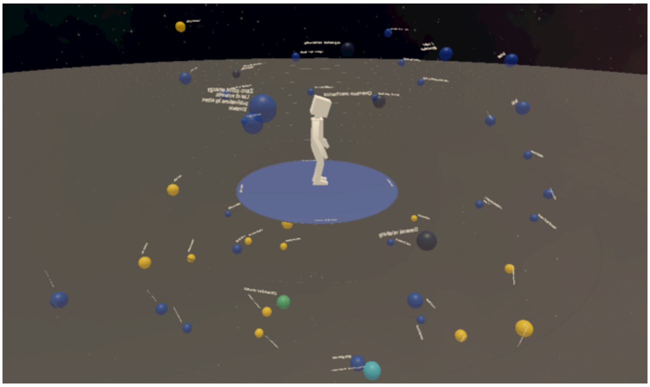


Figure 2. An external view of the visualization shows the main fragment (here Albert Einstein) represented as a colored platform the user is standing on. Related Fragments are represented as colored, labeled spheres. Their arrangement gives clues regarding the fragments' interrelationships.

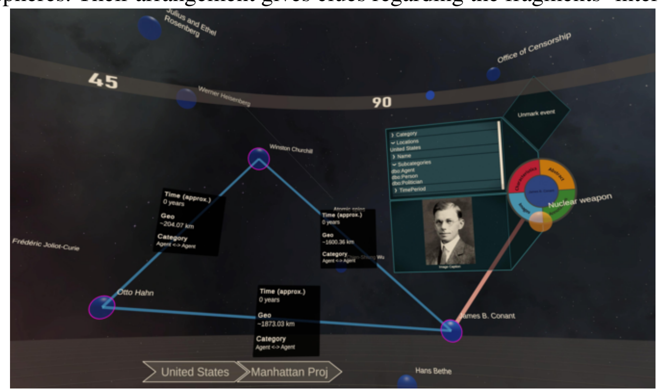


Figure 3. The user explores the scene from an egocentric perspective. For every sphere, its corresponding detail window reveals deeper information on the corresponding fragment by showing up its time period, locations, a picture (if available), and a describing text. The history localized in the lower area of the field of view shows the order of the main fragments visited so far (here United States). In the upper part of the field of view, the compass serves as a guidance line; if nodes are marked, their mark appears on the compass in the colour of their respective category. The feature referred to as 'relation details' helps to figure out how two fragments are related. It shows the gap in time, the shortest distance between their locations, and their categorical differences.

database. The resources of class `dbo:Person`, `dbo:Organization`, `dbo:Species`, `dbo:Language` and `dbo:EthnicGroup` were classified into the category `Agent`. Furthermore, the resources of class `dbo:MeanOfTransportation`, `dbo:Currency`, `dbo:Device`, `dbo:Food` and `dbo:ChemicalSubstance` were classified as `Material Object`. All the resources that could not be classified into one of the above categories are classified into `Miscellaneous`.

In another mapping process, we mapped properties from DBpedia and Wikidata to a minimal set of descriptive properties for each of our seven categories. The descriptive properties are i) `category`, ii) locations, iii) `name`, iv) `sub-categories`, v) `time period` and vi) `properties`. All properties containing geographical information of locations are listed. Respectively, all properties containing time information are collected in the descriptive property `time period`. Table 1 shows an example of six descriptive properties for category `Agent` (i.e., *`res: Albert_ Einstein`*). The descriptive property `properties` consists of 26 properties and some of them are shown in the Table. These sets of minimal descriptive properties of each resource were used in the detail window of a fragment in the Fragmentarium as a standardized brief description (see Figure 3).

	Descriptive properties	**Values**
1	`category`	Agent
2	`name`	Albert Einstein
3	`locations`	Germany, United States, Switzerland, Austria
4	`subcategories`	dbo:Agent, dbo:Person, dbo:Scientist
5	`time period`	0.1.01.1879 - 31.12.1955

6	`properties/gender`
	`properties/positions`
	`properties/religion`
	`properties/abstract`
		male
		professor
		pantheism
		”Albert Einstein was a German-born theoretical physicist, widely acknowledged to be one of the greatest physicists...”

Table 1. An example of six descriptive properties for the resource *res:Albert_Einstein*. The descriptive property `properties` consists of 26 properties for this resource.

To access a knowledge graph database and apply adequate filter functions, an API was developed. The API allows us to query the database for specific resources that can be visualized as historical fragments in the Fragmentarium. In addition, filters could be applied to query the database for resources in certain geographical places or time frames. Additionally, the API was used to compute the central and relative semantic distances between fragments based on the abstracts from DBpedia.

4. User Study

In this section, we present a user study that helped us to gain insights into how users interact with eTaRDiS. The user study was designed with two goals: 1) to evaluate if the user (a historian or a layperson with an interest in history) could retrieve relevant information regarding a certain historical question, 2) to assess the usability of the application.

The participants started in the Hub and were asked to open a preset with the main fragment Hundred Years’ War (i.e., the resource `res:Hundred_Years'_War`) as the starting point in the Fragmentarium and got a brief tutorial on the main functionalities. After the introduction, they were instructed to choose a preset with the fragment Albert Einstein (i.e., the resource `res:Albert_Einstein`) as the starting point of four successive tasks and were asked to “think aloud” (method based on [35]) in the Fragmentarium. In this section, we describe the tasks, participants, and results of the user study.

4.1. Tasks

The underlying question for the tasks was: *‘What is the relation between the fragments `World War II (WWII)`, `J. Robert Oppenheimer (Oppenheimer)` and the `Manhattan Project?`’*. In order to answer the question, the participants were first asked to find the three fragments in the Fragmentarium. As a second task, they had to retrieve information on the location and time for each fragment. As a third task, they were then asked to read the abstract of the fragments and explore the distance to other fragments as a basis to find the relation between the different fragments. Finally, as a fourth task, they were asked to assess *'how prominent the relation between the different fragments was'*. We relied on the System Usability Scale (SUS) developed by Brooke [13] as a basis for a questionnaire administered to the participants, adapting the questions to the eTaRDiS setting. The participants assigned a score on a five-point scale from *'not true at all'* to *'fully agree'* for each of the ten questions regarding aspects of the usability of eTaRDiS.

4.2. Participants

The participants were recruited through a seminar of historical studies and the digital humanities and received credits for their participation in the study. In total, 13 students participated in the user study. The age of the participants was between 24 and 42 years (average age 29.13). Six participants had a background in historical studies. Three participants even had an occupation related to history, e.g., as an archivist, or they were working in a project related to history. Five participants were interested in history and had a background in different research areas of social sciences (see Section 3.1). The interviews were transcribed and analyzed using qualitative content analysis. The total dataset comprises 2 hours 48 minutes and 41 seconds of audio and video recording.

4.3. Results

For eleven out of 13 participants, eTaRDiS was the first VR experience. All participants were able to complete the given tasks and eleven participants found the relation easy to identify. In the first task, the participants needed between 8 and 84 seconds (42.85 seconds on average) to find the three relevant fragments. One participant inferred from the names of persons on fragments coloured in blue that all blue fragments denote Agents and used this correct inference to find the (blue-coloured) fragment for Oppenheimer. Another participant used a sphere's size to find the fragment WWII, assuming the sphere for the event to be relatively large. Two other participants used the labels of the surrounding fragments to locate Oppenheimer and the Manhattan Project, assuming the fragments had to be in the proximity of the fragments of physicists. Four participants marked the corresponding spheres. The proximity of the fragments Manhattan Project and WWII represented a challenge to the exploration for some participants as the label of one of the fragments was partially occluded by the open detail window of other fragments.

In the second task, the participants used three strategies to find the location and time of the fragments. The first strategy used by six participants was to open the characteristics in the detail windows of the fragments and read the information under location and time period. For WWII, no locations were given but the participants found the relevant information in the compilation of the place property. The second strategy used by five participants was to read the abstract of each fragment. The third strategy used by two participants was to read the characteristics and the abstracts. In all strategies, the participants zoomed in or moved the detail windows for a better view of the texts.

In the third task, the participants also used three strategies to identify the topical relation of the fragments. In the first strategy, three participants read the abstracts. In particular, the participants used the names of the fragments as keywords to search for relevant information in the abstracts. In the second strategy, six participants used the relation details in combination with the abstracts, and in the third strategy, three participants additionally opened the images given for the fragments. One participant answered the question without using any of the information or functions in the Fragmentarium.

As stated before, eleven participants out of 13 found the relation of the three fragments easy to identify with the information and functions given in the Fragmentarium. In the third and fourth tasks, six participants stated that they had prior knowledge regarding the relation of the fragments. Five of these six participants were in the group of historians. The layperson with prior knowledge stated that they knew that *"Oppenheimer invented the atomic bomb [and] that it goes hand in hand with the Second World War, of course. However, the fact that the Manhattan Project is related to the atomic bomb and that Oppenheimer also belonged to the Manhattan Project, would not have come to my mind before"*. Several information and functions were seen as helpful by the participants to identify the relation. One participant referred to the images given for a fragment, and one participant referred to the proximity of the fragments. Two participants referred to the relation details, the relative distance weighted by category, or related fragments. Four participants found the abstract and other textual information helpful.

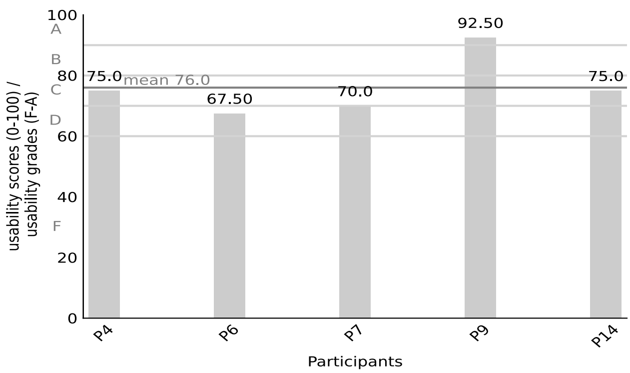


Figure 4. SUS scores of laypersons.

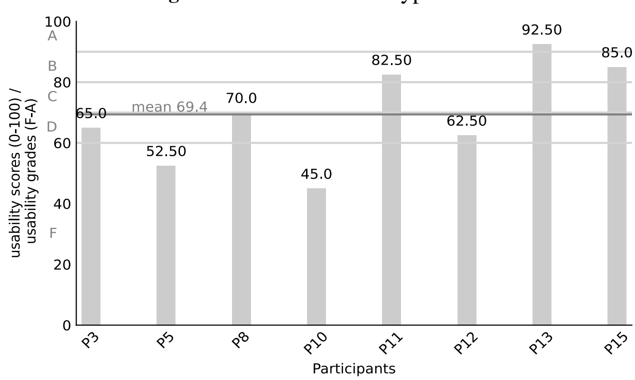


Figure 5. SUS scores of historians.

In the evaluation of the usability questionnaire, we computed the SUS score for each participant, the mean overall, and the mean for each of the two groups, laypersons (five participants) and historians (eight participants) (see Figure 4). We use the scheme of Bangor et al. [36] as a reference for the interpretation of the resulting SUS scores regarding the acceptability of the system. In addition to the usability scale, we give the corresponding school grades for the system suggested by Bangor et al. in Figure 5. The overall mean SUS score is 71.92, which corresponds to a ‘satisfactory’ rating in the interpretation scheme of Bangor et al. However, the mean of SUS scores is different for the two groups of participants. The mean score of laypersons interested in history is 76.0, corresponding to a ‘satisfactory to excellent’ rating. The mean score of historians is 69.4, corresponding to a ‘sufficient to satisfactory’ rating.

5. Discussion

Most of the participants were able to use the functionalities of the virtual environment without any problems. They used the existing features and tested the boundaries of the rules for the interaction with objects in the virtual environment, e.g., when they tried to use the grab button to move the fragments or other fixed elements (e.g., the windows of the relation details). Some participants who used a VR application for the first time stated that they liked the playful approach to exploring historical data. Two participants who worked in an educational context suggested using eTaRDiS for knowledge transfer. Additionally, the participants stated that they learned something new. For example, one participant said *“didn’t know the topic before, but that the connection exists, you could actually find out pretty quickly by reading the abstract”*. Three participants even looked for information unrelated to the task. One participant stated regarding the information in the characteristics of the fragment WWII: *“Yes, that is also interesting. How many casualties have there been?”* (usability study 2023, P9). Another participant looked at the other fragments related to Albert Einstein and stated: *“See what else is around [the fragment WWII]. Physical review. Munich. Mozart is irritating me a bit. Why is Mozart there?”* (usability study 2023, P4). A general observation was that the participants tried to explain the proximity of fragments or why fragments were visualized in the Fragmentarium with their prior historical knowledge. Therefore, the VR environment stimulated the reflection of their prior knowledge and the potential generation of new knowledge by discovering new relationships between historical fragments.

On the other hand, some participants commented after participating in the study they felt overwhelmed by the novel visual impressions in eTaRDiS due to their lack of VR experience. However, in the usability questionnaire, four participants gave five out of five points (corresponding to the answer ‘fully agree’) and six participants gave four out of five points for the question if they would like to use eTaRDiS more often. The other three participants gave three out of five points, corresponding to a neutral answer.

eTaRDiS allows users to take different perspectives on a topic, and the results showed that the participants found the implemented features helpful for this. In particular, the participants explored the adjacency of individual fragments to put them into context. In addition, they combined different aspects like the arrangement of the fragments, their detail windows, and their relation details.

The SUS scores of 76.0 (mean) for the group of laypersons corresponding to a ‘satisfactory to excellent’ rating and the mean score of historians of 69.4 corresponding to a ‘sufficient to satisfactory’ rating are a promising first rating for our prototype system.

Another perspective on the SUS scores is gained by considering the participant's digital media affinity (DMA), which users were asked to indicate in a self-assessment on a six-point scale, in addition to their prior VR experience. Our DMA scale ranges from *'very bad'* to *'extremely good'*. Nine participants who stated that they had a *'good'* to *'extremely good'* DMA have a SUS score between 62.5 and 92.5 (avg. 77.8) and four participants who rated themselves as *'very bad'* to *'not bad'* at using digital media have a SUS score between 45.0 and 70.0 (avg. 58.8). An independent samples t-test between the groups yields a p-value of 0.039 and an absolute t-value of 2.70, which shows a high influence of the stated DMA on the SUS score. We observed a positive Pearson correlation between DMA and SUS scores, r = 0.75. Based on the threshold values of Cohen[37], these effect size can be considered large. Two participants stated that they are regularly using VR applications. Their SUS scores are 70.0 corresponding to a 'satisfactory' rating and 82.5 corresponding to an 'excellent' rating. Eleven participants had no VR experience and had SUS scores between 45.0 and 92.5 (avg. 71.1). The broad range of SUS scores indicates that the lack of VR experience might not be of relevance regarding perceived usability. The p-value of 0.57 in an independent samples t-test for the SUS scores in the two groups with or without VR experience shows that the difference in the two groups is not statistically significant. Therefore, the quite high results in the group with VR experience can be seen as informed trends but need to be validated in further studies.

6. Conclusion and Future Work

We presented eTaRDiS, a VR exploration tool to support access to knowledge graphs for historians. In the presented user study, our tool achieved an overall SUS score of 71.92, corresponding to a 'satisfactory' rating. While the mean score reached with laypersons interested in history was quite high with 76.0, corresponding to a rating of 'satisfactory to excellent', the score for historians was lower with 69.4, corresponding to a 'sufficient to satisfactory' rating. Further analysis showed that participants who rated themselves as *'good'* to *'extremely good'* at using digital media have a higher mean SUS score of 77.8 compared to the participants who rated themselves as *'very bad'* to *'not bad'* at using digital media (mean SUS score: 58.8). The findings of the usability evaluation and the qualitative analysis of exploration patterns show the system's potential to be a valuable tool for allowing digital humanities researchers and laypersons to explore knowledge graphs. With eTaRDiS, such users are able to reflect on their prior knowledge as well as generate new knowledge by discovering new relationships between historical fragments.

In future work, we aim to integrate other historical databases to allow more diverse data to be explored using eTaRDiS. In addition, the feedback of the participants in the user study provided insights that we aim to integrate into eTaRDiS via further features specific to research purposes in historical studies.

Acknowledgements

This research is part of the project eTaRDiS[8] (Exploration of Temporal and Spatial Data in Immersive Scenarios), which is funded by the Federal Ministry of Education and Research (BMBF). Furthermore, we are grateful to all participants attending our user study.

[8] https://digital-history.uni-bielefeld.de/etardis/

References

[1] Hyvönen E. Cultural Heritage on the Semantic Web. In: Publishing and Using Cultural Heritage Linked Data on the Semantic Web. Cham: Springer International Publishing; 2012. p. 1-11. doi:10.1007/978-3-031-79438-4_1.

[2] Hyvönen E. "Sampo" model and semantic portals for digital humanities on the Semantic Web. In: DHN 2020 Digital Humanities in the Nordic Countries. Proceedings of the Digital Humanities in the Nordic Countries 5th Conference, CEUR Workshop Proceedings. vol. 2612; 2020. p. 373-8.

[3] Nünning V, Nünning A, Neumann B, Horn M. Cultural Ways of Worldmaking: Media and Narratives. Berlin, New York: De Gruyter; 2010. doi:doi:10.1515/9783110227567.189.

[4] Karadimas D, Somakos L, Bakalbasis D, Prassas A, Adamopoulou K, Karadimas G. Current and Potential Applications of AR/VR Technologies in Cultural Heritage. "INCEPTION Virtual Museum HAMH: A Use Case on BIM and AR/VR Modelling for the Historical Archive Museum of Hydra Greece". In: Moropoulou A, Korres M, Georgopoulos A, Spyrakos C, Mouzakis C, editors. Transdisciplinary Multispectral Modeling and Cooperation for the Preservation of Cultural Heritage. Cham: Springer International Publishing; 2019. p. 372-81.

[5] Skamantzari M, Kontogianni G, Georgopoulos A, Kazanis S. Developing a virtual museum for the Stoa of Attalos. In: 2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games); 2017. p. 260-3. doi:10.1109/VS-GAMES.2017.8056611.

[6] McCaffery J, Miller A, Vermehren A, Fabola A. The Virtual Museums of Caen: A case study on modes of representation of digital historical content. In: 2015 Digital Heritage. vol. 2; 2015. p. 541-8. doi:10.1109/DigitalHeritage.2015.7419571.

[7] Józsa O, Börcs A, Benedek C. Towards 4D virtual city reconstruction from Lidar point cloud sequences. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. 2013 05;II-3/W1:15-20. doi:10.5194/isprsannals-II-3-W1-15-2013.

[8] Ayers EL. Turning Toward Place, Space, and Time. In: David J B, John C, Trevor M H, editors. The Spatial Humanities : GIS and the Future of Humanities Scholarship. Bloomington & Indianapolis: Indiana University Press; 2010. p. 1-13.

[9] Nünning A. Making Events – Making Stories – Making Worlds: Ways of Worldmaking from a Narratological Point of View. In: Nünning V, Nünning A, Neumann B, Horn M, editors. Cultural Ways of Worldmaking: Media and Narratives. Berlin, New York: De Gruyter; 2010. p. 189-214. doi:doi:10.1515/9783110227567.189.

[10] Eakin PJ. How Our Lives Become Stories: Making Selves. Cornell University Press; 1999.

[11] Sicart M. Defining Game Mechanics. Game Studies. 2008;8(2).

[12] Derksen M, Weissker T, Kuhlen T, Botsch M. Towards Discovering Meaningful Historical Relationships in Virtual Reality. In: 2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW); 2023. p. 697-8. doi:10.1109/VRW58643.2023.00191.

[13] Brooke J. SUS-A quick and dirty usability scale. In: Usability evaluation in industry. 1st ed. London: CRC Press; 1996. p. 189-94. doi:10.1201/9781498710411.

[14] Pirch S, Müller F, Iofinova E, Pazmandi J, Hütter CVR, Chiettini M, et al. The VRNetzer platform enables interactive network analysis in Virtual Reality. Nature Communications. 2021;12(1):24-32. doi:10.1038/s41467-021-22570-w.

[15] Buphamalai P, Kokotovic T, Nagy V, Menche J. Network analysis reveals rare disease signatures across multiple levels of biological organization. Nature Communications. 2021;12(1):6306. doi:10.1038/s41467-021-26674-1.

[16] Han D, Parsad G, Kim H, Shim J, Kwon OS, Son KA, et al. HisVA: A Visual Analytics System for Studying History. IEEE Transactions on Visualization and Computer Graphics. 2022;28(12). doi:10.1109/TVCG.2021.3086414.

[17] Cho I, Dou W, Wang DX, Sauda E, Ribarsky W. VAiRoma: A Visual Analytics System for Making Sense of Places, Times, and Events in Roman History. IEEE Transactions on Visualization and Computer Graphics. 2016;22(1):210-9. doi:10.1109/TVCG.2015.2467971.

[18] Krishnan M, Ober J, Pyzyk M. POLIS : Designing a Visualization Tool for the Research of Complex Sociopolitical Landscapes. Parsons Journal for Information Mapping (PJIM). 2014;6(2):1-9.

[19] Kidd J. "Immersive" heritage encounters. The Museum Review. 2018;3(1).

[20] Immersive Virtual Reality. In: Furht B, editor. Encyclopedia of Multimedia. Boston, MA: Springer US; 2008. p. 345-6. doi:10.1007/978-0-387-78414-4_85.

[21] Casu A, Spano LD, Sorrentino F, Scateni R. RiftArt: Bringing Masterpieces in the Classroom through Immersive Virtual Reality. In: Giachetti A, Biasotti S, Tarini M, editors. Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference. The Eurographics Association; 2015. p. 77-84. doi:10.2312/stag.20151294.

[22] Razum M, Sack H, Tietz T, Bruns O, Göller S, Scharm T, et al. TOPORAZ: Ein digitales Raum-Zeit-Modell für vernetzte Forschung am Beispiel Nürnberg. Information - Wissenschaft & Praxis. 2020;71(4):185-94.

[23] McIntire JP, Liggett KK. The (Possible) Utility of Stereoscopic 3D Displays for Information Visualization: The Good, the Bad, and the Ugly. In: 2014 IEEE VIS International Workshop on 3DVis (3DVis); 2014. p. 1-9. doi:10.1109/3DVis.2014.7160093.

[24] Wagner Filho JA, Rey MF, Freitas CMDS, Nedel L. Immersive Visualization of Abstract Information: An Evaluation on Dimensionally-Reduced Data Scatterplots. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR); 2018. p. 483-90. doi:10.1109/VR.2018.8447558.

[25] Kraus M, Weiler N, Oelke D, Kehrer J, Keim DA, Fuchs J. The Impact of Immersion on Cluster Identification Tasks. IEEE Transactions on Visualization and Computer Graphics. 2020;26(1):525-35. doi:10.1109/TVCG.2019.2934395.

[26] Wagner J, Stuerzlinger W, Nedel L. The Effect of Exploration Mode and Frame of Reference in Immersive Analytics. IEEE Transactions on Visualization and Computer Graphics. 2022;28(9):3252-64. doi:10.1109/TVCG.2021.3060666.

[27] McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software. 2018;3(29):861. doi:10.21105/joss.00861.

[28] Clark E, Celikyilmaz A, Smith NA. Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics; 2019. p. 2748-60. doi:10.18653/v1/P19-1264.

[29] 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). Hong Kong, China: Association for Computational Linguistics; 2019. p. 3982-92. doi:10.18653/v1/D19-1410.

[30] Unsworth J. Scholarly primitives: What methods do humanities researchers have in common, and how might our tools reflect this. In: Symposium on Humanities Computing: Formal Methods, Experimental Practice, King's College. London; 2000. .

[31] Pacheco A. Digital humanities or humanities in digital: revisiting scholarly primitives1. Digital Scholarship in the Humanities. 2022 03;37(4):1128-40. doi:10.1093/llc/fqac012.

[32] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web. 2015;6(2):167-95.

[33] Vaidya G, Kontokostas D, Knuth M, Lehmann J, Hellmann S. DBpedia Commons: Structured Multimedia Metadata from the Wikimedia Commons. In: Proceedings of the 14th International Semantic Web Conference; 2015. .

[34] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of the 6th International Semantic Web Conference (ISWC). vol. 4825 of Lecture Notes in Computer Science. Springer; 2008. p. 722-35.

[35] Van Someren M, Barnard YF, Sandberg J. The think aloud method: a practical approach to modelling cognitive processes. London: AcademicPress. 1994;11.

[36] Bangor A, Kortum P, Miller J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. Journal of Usability Studies. 2009 may;4(3):114–123. doi:10.5555/2835587.2835589.

[37] Cohen J. Statistical Power Analysis for the Behavioral Sciences. 2nd ed. New York: Routledge; 1988. doi:10.4324/9780203771587.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230012

TRANSRAZ Data Model: Towards a Geosocial Representation of Historical Cities

Oleksandra BRUNS [a,b], Tabea TIETZ [a,b], Sandra GÖLLER [a] and Harald SACK [a,b]
[a] *FIZ Karlsruhe – Leibniz Institute for Information Infrastructure*
firstname.lastname@fiz-karlsruhe.de
[b] *Karlsruhe Institute of Technology (AIFB)*

Abstract.
Preserving historical city architectures and making them (publicly) available has emerged as an important field of the cultural heritage and digital humanities research domain. In this context, the TRANSRAZ project is creating an interactive 3D environment of the historical city of Nuremberg which spans over different periods of time. Next to the exploration of the city's historical architecture, TRANSRAZ is also integrating information about its inhabitants, organizations, and important events, which are extracted from historical documents semi-automatically. Knowledge Graphs have proven useful and valuable to integrate and enrich these heterogeneous data. However, this task also comes with versatile data modeling challenges. This paper contributes the TRANSRAZ data model, which integrates agents, architectural objects, events, and historical documents into the 3D research environment by means of ontologies. Goal is to explore Nuremberg's multifaceted past in different time layers in the context of its architectural, social, economical, and cultural developments.

Keywords. cultural heritage, digital humanities, city exploration, knowledge graphs, archival documents, architecture

1. Introduction

City preservation plays an important role in the cultural heritage and digital humanities research domains. Exploring the historical development of city architectures along with people living in it, their progress in technology, their craftsmanship as well as arts and culture is highly relevant for historians, architects, sociologists and the general public. Modernization, economic and industrial development, environmental progress, and the occurrence of natural disasters and wars result in the ongoing transformation of the city. Without a digital preservation of culturally relevant locations and the curation of their historical resources, this important heritage will be lost forever. In recent years, a number of projects have dealt with the digital preservation of culturally meaningful cities and places[1] [22]. In line with these efforts is the project TRANSRAZ[2], a successor of the

[1] https://www.timemachine.eu/
[2] https://www.fiz-karlsruhe.de/en/forschung/transraz

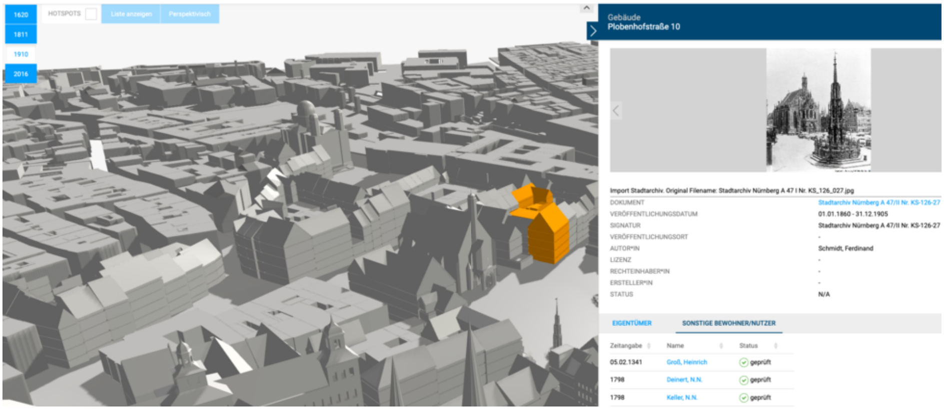


Figure 1. 3D visualisation of the building located on Plobenhofstraße 10 in 1910, and additional information about the building in the TRANSRAZ VRE.

project TOPORAZ [24]. Within TRANSRAZ heterogeneous historical data collections about persons, organizations and events are connected to an architectural 3D virtual research environment (VRE) using Knowledge Graphs (KGs) to enable the exploration of the historic city of Nuremberg in different time periods ranging from the Middle Ages to the 21st century. Nuremberg was one of the great European metropolises in the Middle Ages and beyond. It was the birthplace of renaissance artist Albrecht Dürer, who worked there all his life. The city developed into the epitome of German and European history and culture. Then, during the Second World War, the city was largely destroyed and only few buildings could be reconstructed. Therefore, a systematic and scientific reconstruction of the city in different time periods is necessary to research the history of Nuremberg in the context of its architectural, social, economical, and cultural developments.

In the TRANSRAZ VRE, KGs are utilized to connect historical documents retrieved by domain experts from archives and museums along with the entities extracted from the documents with the architectural 3D model (see Figure 1). These entities include among others persons, events, organizations and occupations. KGs furthermore allow to enrich these with additional information from external resources like Wikidata[3] and the German Authority Files[4] (GND). The VRE will allow to research residential and working areas, the establishment of educational institutions over time, the development of industry and the distribution of wealth in the city[5]. Furthermore, the connection of historical sources with the VRE allows for a scientifically accurate exploration. KGs have proven useful and valuable to integrate and enrich these heterogeneous data. However, this task also comes with a number of data modeling challenges, which include mapping a relational data model to existing ontologies, the efficient representation and management of data provenance, the change of entities over time, and the connection to the 3D environment.

The contribution of this paper is the TRANSRAZ data model, which includes ontologies to represent historical persons, their occupation and addresses, events, document

[3] https://www.wikidata.org/wiki/Wikidata:Main_Page
[4] https://www.dnb.de/EN/Professionell/Standardisierung/GND/gnd_node.html
[5] https://www.toporaz.de/toporaz-current/explore

annotations, and buildings in a 3D research environment. The TRANSRAZ data model consists of four main building blocks which have to be integrated into the VRE.

1. **Architectural Objects** are modeled as 3D elements within the VRE. The building parts are created as separate elements and have to be identifiable uniquely.
2. **Historical documents** are retrieved from archives and museums and contain relevant historical information about cities inhabitants, events etc. By means of information extraction and semi-automated annotations, the documents are connected to the 3D architecture with respect to their specific location and time mentioned.
3. **Agents** can be inhabitants of the historical city of Nuremberg and organizations. Entities, along with (family) relations, dates, addresses, and occupations are extracted from the existing TOPORAZ database as well as the historical document collections and connected to the 3D environment.
4. **Events** are extracted from the existing TOPORAZ database as well as the historical document collections. They include important city events and festivals as well as changes of state, e.g. architectural changes.

Goal is to represent these four main building blocks in a semantically meaningful and efficient way to enable a scientifically correct and intuitive exploration of Nuremberg in the context of its inhabitants and important events on the foundation of historical documents.The generalization of this contribution is ensured as it can be applied to any project related to the semantic representation of historical entities in a location based exploration environment. This work is reusable and reproducible, the data model is publicly available on the Web[6] including its documentation.
The remainder of the paper is structured as follows. In section 2 works related to modeling architectural data, the representation of heterogeneous cultural heritage data and the alignment of archival documents are discussed. In section 4 the design methodology along with data ressources and competency questions is discribed. Section 3 contains the main contribution of the paper, followed by evaluation use cases to explore the TRANSRAZ data. Section 5 concludes this paper.

2. Related Work

Three-dimensional models have traditionally been the purview for manufactures of navigation systems, planners, pollution researchers and virtual tourists [2]. However, in recent years the interest of connecting modern digital technologies with historical data from the past has raised immensely. Historians, museologists, architects and the general public are keen to research and curate heritage data, and basically to walk through the history hidden inside historical materials. Adding new dimensions to the past is a goal of the Time Machine projects[7]. Discovery, extraction, connection, reuse of historical data, and subsequently, the 3D reconstruction of European cities, e.g. Amsterdam[8] or Venice[9], is an ongoing goal to provide users with the ability to travel back in time. Meanwhile, a 4D browser for researching and communicating the history of the city of Dresden is

[6] https://ise-fizkarlsruhe.github.io/Transraz/datamodel
[7] https://www.timemachine.eu/about-us/
[8] https://www.amsterdamtimemachine.nl/
[9] https://www.epfl.ch/research/domains/venice-time-machine/

described in [22]. Apart from urban spaces, several projects aim at a detailed reconstruction of distinguished architectural objects. For example, Florence4D[10] is an initiative to revive prominent buildings of the Renaissance Florence. Similarly, the project ”Digital Reconstruction of the Breslau Synagogue”[11] aims at reconstructing the largest synagogue in Breslau. A semantic information model will be presented, connecting objects with the resources used for the reconstruction. Additionally, the synagogue will be put into the social context by linking it to persons, corporate bodies, historical events, etc. Furthermore, there are several projects that utilize open source HBIM (Historic Building Information Modeling) software[12][7], including the HBIM Wiki Project[13], which aims to create an open database of historical buildings, the OpenHeritage project[14] that documents and conserves historic buildings across Europe, the H-BIM platform[15] and the Arches-HBIM project[16], which integrate HBIM with heritage management systems. In the context of these efforts, TRANSRAZ aims at not only 3D reconstructing of the entire urban environment of Nuremberg in 1910, but at enhancing the VRE via connecting its architectural objects to knowledge from external resources. To enable the enrichment, interoperability and reuse, tangible (e.g., historical buildings, places, documents) and intangible (e.g., festive events, traditions, life lines of persons) cultural heritage is meaningfully represented my means of ontologies.

Cultural heritage objects vary significantly in media type and attributes, however, they still share semantic similarities and benefit from cross-connections. The last decades have shown that semantic interlinking and representation of heterogeneous cultural heritage data is of interest for many research projects. There have been attempts to semantically model the data of Korean [21], Italian [6] and Finnish [18] national heritage. As part of the Europeana Project, the European Data Model (EDM) [9] was developed to provide a shared ontology infrastructure for cultural data from European GLAM institutions. All these data models are similar in nature of modeling, since the main reference for the models is the Conceptual Model CIDOC-CRM [8], a domain ontology of cultural heritage. Recently German National Library (DDB) has launched an initiative to develop the DDB Knowledge Graph and, thus, to enhance the frontend of the DDB. This is due to the complicated modeling of the EDM that negatively influences the usability and exploration of the data [25]. Despite CIDOC-CRM being also not fully sufficient for the user-oriented purposes of TRANSRAZ data model, for reasons of interoperability, the TRANSRAZ data model follows the best practices and provides a mapping to CIDOC-CRM.

A challenge of the modeling of TRANSRAZ space is the heterogeneity of the cultural heritage data that is obtained from a great amount of various resources, and that has to be represented in a direct connection with each other. This challenge can be addressed by exploiting existing ontologies and vocabularies. Related work towards modeling specific building blocks and design choices is discussed in section 3.2.

[10] https://florence4d.org/s/florence4d/page/home
[11] https://architekturinstitut.hs-mainz.de/projekte/digitale-rekonstruktion-der-breslauer-synagoge/
[12] https://github.com/UNIFE/hbim-suite
[13] http://www.hbimwiki.org/
[14] https://openheritage.eu/
[15] http://www.h-bim.com/
[16] https://www.arches-hbim.org/

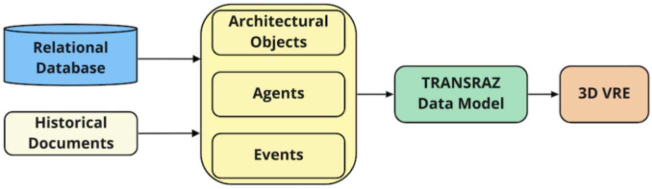


Figure 2. Workflow

3. TRANSRAZ Data Model

This section presents the main contribution of this work, the TRANSRAZ data model, which integrates agents, architectural objects, events, and historical documents into a 3D VRE by means of ontologies. This task presents versatile data modeling challenges: (i) Within TOPORAZ [24], a relational database model and a first version of the VRE were created. A challenge in this ongoing project TRANSRAZ is the transformation of the existing model to a semantic data model with mappings to existing ontologies relevant for the cultural heritage domain, as e.g. CIDOC-CRM. (ii) The historical documents retrieved from archives and museums are analyzed, semi-automatically annotated and connected to the VRE. To allow for a scientifically correct exploration of the city, the document annotation process has to be made completely transparent, i.e. data provenance plays an important role in the modeling decisions. (iii) Within the VRE, users will be able to explore different time periods. That means entities change their names, houses are destroyed and rebuilt, organizations change their ownership etc., which has to be considered in the modeling process.

The overall workflow is shown in figure 2. Architectural objects, agents, and events are represented based on ontologies on the foundation of the existing TOPORAZ relational database and the analyzed historical documents obtained by domain experts. They are then integrated into the TRANSRAZ data model and finally the 3D VRE.

In the following, the building blocks of the TRANSRAZ data model, the modeling requirements for each building block as well as the resulting ontologies are described. Additionally, the evaluation of the data model with the CQs is provided.

3.1. From Data Sources to Building Blocks

Due to the complexity, diversity and heterogeneity of the data sources and the knowledge contained within, a modular modeling approach to construct the data model was adopted. Therefore, the modules are represented by four main building blocks that correspond to the different aspects of the data model. Accordingly, the TRANSRAZ data model is organized in the following blocks:

* **Historical Documents** are used for the provenance annotation of statements, and also as a basis for automatic semantic annotation with references to other building blocks in the data model and classes in the ontologies.
* **Architectural Objects** represents all structures in the urban space, such as streets, districts, buildings, facades, roofs, rooms, etc.
* **Agents** refers to persons, clubs, organisations, associations, etc., who lived, worked or owned properties in the city or are socially connected to them.

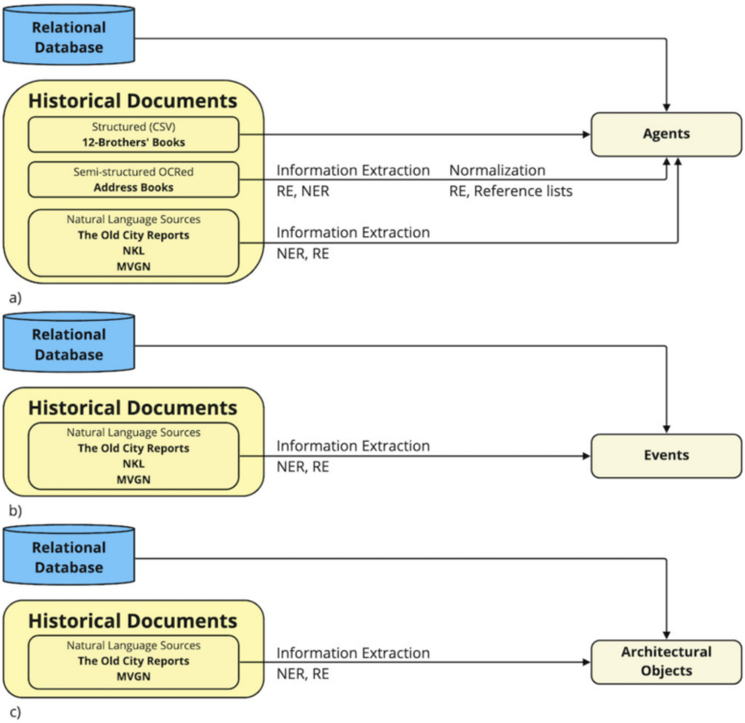


Figure 3. From data sources to building blocks

* **Events** represents events that can be associated with a point in time. On the one hand, this includes social, cultural, political, etc. activities of Nuremberg, such as Oktoberfest 1900. On the other hand, events that represent some duration and/or transformation, as e.g., deconstruction of a building, renaming of a street, etc.

Goals, challenges, and the proposed workflow towards extracting entities from historical resources, linking and disambiguating them are described in [26,5]. Subsequently, in [4], the first results to automatically obtain and model the information in the address books have been presented. Figure 3 schematically shows the workflow to populate the building blocks. However, the detailed discussion and results are out of the scope of this paper and will be addressed in the future work.

3.2. Ontologies of the TRANSRAZ data model

Following the guidelines of the ontology development, the TRANSRAZ data model aims at integrating, reusing, extending or mapping to established vocabularies and ontologies for the representation of the required classes and properties. This section presents concrete ontology engineering requirements that are derived from the CQs discussed in Section 4.2, taking into account the heterogeneity of data and concepts in the desired data model. Additionally, best practices for modeling of each building block are analyzed and compared against the requirements, and the design choices are presented.

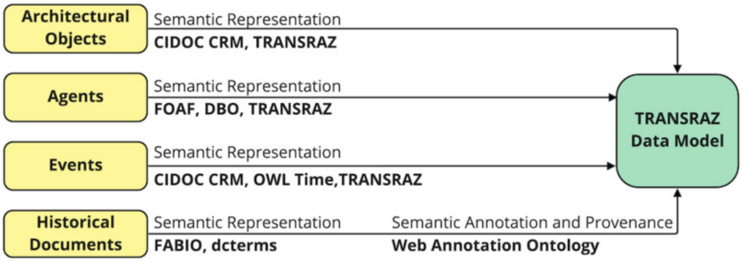


Figure 4. Ontologies of the TRANSRAZ Data Model

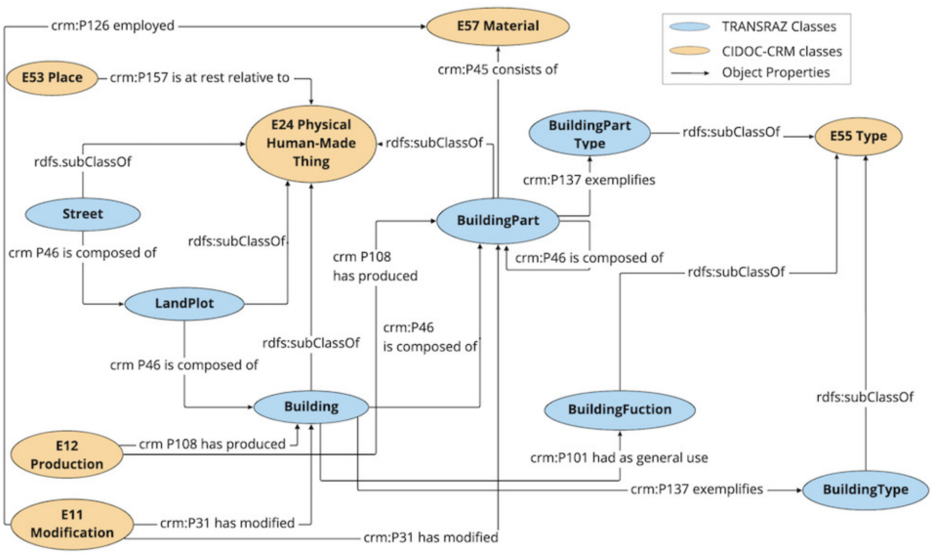


Figure 5. A part of the architectural object modeling within the TRANSRAZ Data Model

3.2.1. *Modeling of City Architecture*

The TOPORAZ relational database is the rich resource of the urban entities of the Main Market of Nuremberg. However, the representation of these entities and their relations is diverse and incomplete. In order to provide a comprehensive semantic representation of the tables in the TOPORAZ data model and further sources described in section 4.1, the following requirements have to be addressed:

REQ1: Architectural objects of a city are represented hierarchically. For example, a city consists of districts, a building consists of floors. This representation may include dozens of levels, describing every small part of a building (CQ4.1-2).

REQ2: Attributes of architectural objects rely on historical resources, thus, may be uncertain or incomplete. For example, for some buildings information about exact height, roof shapes, wall openings (windows, doors) and stylistic elements of the facade are provided, while for other buildings only the location area is known.

REQ3: Architectural objects, their functions, names are highly dynamic and change over time, for example, buildings are destroyed, streets change their names, a house that used to be a pharmacy can become a store, etc (CQ3.2, CQ4.2-3).
REQ4: Due to information dynamics, uncertainty and incompleteness of the data, facts require provenance annotation (CQ6.1-2, CQ6.4).
REQ5: Architectural objects of old cities, such as Nuremberg, are also considered cultural heritage that can further be explored via interconnecting with other cultural heritage objects (CQ3.2, CQ4.3, CQ5.1).

Research has been conducted towards the semantic representation of urban spaces [1,10,20], however, these ontologies are different from the architectural object block of the TRANSRAZ data model in their main goal – they are used to describe a modern city architecture. Moreover, these ontologies lack classes and properties to represent parts of the buildings (REQ1), and thus, are insufficient for modeling the level of detail required for TRANSRAZ buildings. Finally, the ontologies are not flexible and do not consider dynamics (REQ3), because they address urban spaces in real time only.

Due to the lack of ontologies to describe historical cities, a derivation of CIDOC-CRM is provided with CIDOC-CRM being the core ontology of the block. CIDOC-CRM is event-based, which allows the representation of change (REQ3) through events. Also, it provides a vocabulary to represent interconnections among objects and events in a flexible way (REQ2). And finally, it is frequently used by GLAM institutions (galleries, libraries, archives, and museums) to represent diverse cultural heritage, which puts architectural heritage into context with the related cultural objects (REQ5).

Since CIDOC-CRM does not provide an expressive semantic for modeling the structures of buildings and their parts, the model has been enhanced with the addition of self-defined classes, as e.g. *transraz:BuildingPart*. For readability, figure 5 reports only part of the elements of the TRANSRAZ building block, the full modeling is provided on GitHub[17]. The information is organized around three main classes:

* Class *crm: E24 Physical Human Made-Man Thing* allows to define subclasses of the architectural hierarchy and describe their components.
* Class *crm:E55 Type* connects architectural objects and their parts with functions, e.g. library, bakery or sleeping room, or their types, e.g. front house, park, balcony. Both functions and types are completed by a set of controlled vocabularies developed by the domain experts during the TOPORAZ project.
* Class *crm:E5 Event* and its subclasses, e.g. *crm:E12 Production* and *crm:E11 Modification*, provide a description of building processes. For instance, when a building was built or transformed, who did it and what materials were used.

TRANSRAZ classes are defined as subclasses of CIDOC-CRM and, thus, inherit all properties to specify the time of an event, the hierarchical nature of architectural objects, their types, functions, etc. Moreover, all architectural objects are associated with geographical coordinates and addresses that work as a bridge to connect them to the agents.

3.2.2. Modeling of Historical Agents.

One of the main goals of the TRANSRAZ project is to depict historical Nuremberg not only geographically, but also in its social context. Thus, the semantic representation of

[17] `https://github.com/ISE-FIZKarlsruhe/Transraz/`

its citizens and companies and their social networks is necessary. After the analysis of the complex and heterogeneous resources that contain information about persons and organizations of Nuremberg (see figure 3(a)), the following modeling requirements have been identified:

REQ1: Agents of a city are interconnected via different relations, which are of great interest to historic and genealogical research and have to be semantically represented (CQ1.1, CQ2.1, CQ2.3, CQ5.2).

REQ2: The descriptions of agents are often extracted from the historical resources, thus, may be uncertain, incomplete or even false. Thus, provenance annotation is required (CQ6.1-4).

REQ3: Attributes of agents, e.g. their addresses, names, occupations, etc. may change overtime (CQ1.1, CQ2.1-2, CQ3.1, CQ5.2).

There exist several ontologies that specialize in description of agents from different perspectives. However, they are mostly focused on describing either their existence in the virtual world, e.g. their activities on the Web [12], or basic information required for postal delivery [19]. In this work, agents serve as linking points among the building blocks, hence, among different domain-specific ontologies. The DBpedia Ontology (DBO) benefits from linking to the most common ontologies, as e.g. SCHEMA.ORG [18], Wikidata [19], FOAF [20]. Thus, the properties and classes from different vocabularies are interconnected and could simply be reused. Additionally, DBO contains a rich semantic representation and hierarchy of social relations, e.g. *dbo:mother, dbo:sister, dbo:friend* (REQ1). Mapping to CIDOC-CRM classes and properties, e.g. to CRM's *E21 Person*, enables the direct use of the CIDOC-CRM's events, that allow for the representation of change (REQ 3) (see figure 6).

3.2.3. Modeling of Events and Changes.

In historical research, when working with cultural data it is important to observe the development of entities and to keep track of changes that occur over time. That is why a proper semantic representation of temporal happenings is essential for the TRANSRAZ data model. Based on the data and data source analyses, the following requirements for modeling events have been established:

REQ1: Events in the TRANSRAZ model represent both social activities and changes of state. For example, the premier of the "Zapfenstreich" in 1907 and the change of address for a person or deconstruction of a building (CQ1.1, CQ2.4, CQ3.1-2, CQ4.2-3).

REQ2: Events of state change cover both semantic (occupation changes its function over time) and terminological (street is renamed) change of entities (CQ1.1, CQ2.2, CQ4.3).

REQ3: Events can be continuous and discreet, and thus, associated with time points or time intervals.

[18] https://schema.org/
[19] https://www.wikidata.org/wiki/
[20] http://xmlns.com/foaf/spec/

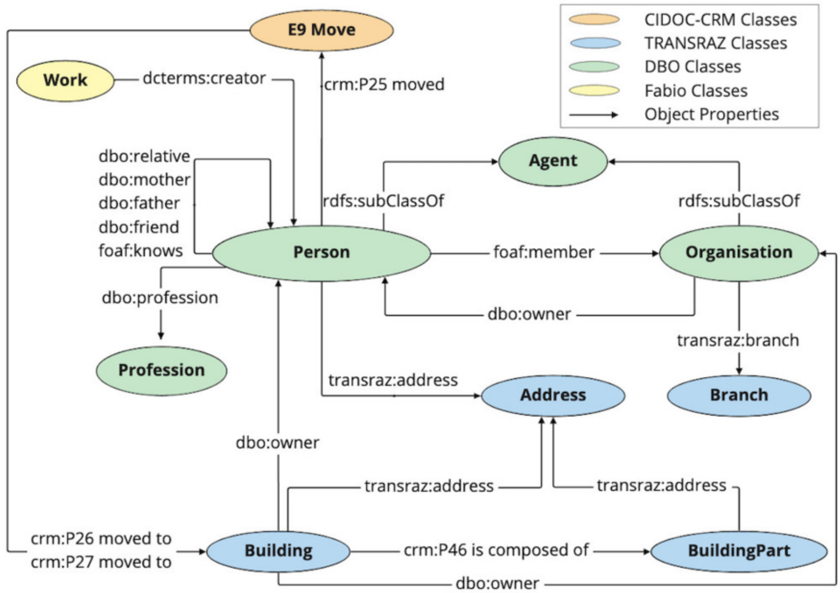


Figure 6. A part of the modeling of the TRANSRAZ agents block

REQ4: Events and their attributes, e.g. temporal information, are often extracted from the historical resources, thus, may be uncertain, incomplete or even false. Thus, provenance annotation is required (CQ6.1, CQ6.2, CQ6.3, CQ6.4).

REQ5: Events can be interconnected via temporal relations, as e.g., *overlaps, before* (CQ1.1, CQ2.1-2, CQ3.1-2, CQ4.2).

The entity-based representation of temporal relations requires either additional extensions to RDF [14,27,15], or a great amount of additional statements [17], while an event-based approach addresses this gap and allows for direct representation. In [11], a temporal event-centric Knowledge Graph is presented. In the work, subclasses of Wikidata and DBpedia classes *wdt:Q1656682* (Event) and *dbo:Event* are identified. However, these ontologies model events that aim at representing certain activities, e.g. lecture, concert, festival, and do not cover events that represent temporal states and change (REQ1, REQ2). The EventKG enhances the Simple Event Model (SEM) [28] and provides a vocabulary for modeling temporal relations of the types entity-entity (e.g. marriage), entity-event (e.g. Margarete Haagen played the Girl in the "Zapfenstreich"), but do not directly cover temporal relations between events, e.g. does reconstruction of the "Frauenkirche" overlap with the reconstruction of the St. Lorenz Church? (REQ5).

In contrast, CIDOC-CRM provides semantics for changes of state in cultural, social and physical systems, e.g. class *crm:E11 Modification* and its subclasses comprise a change of an object, class *crm:E8 Acquisition* comprises transfers of legal ownership from one agent to another (REQ1). Moreover, CIDOC-CRM's events are represented through the time spans of their validity in form of abstract temporal extents (REQ3), e.g. duration of the Ming Dynasty, and are connected to each other via temporal relations (REQ5), e.g. *crm:P114 is equal in time to, crm:P117 occurs during*, etc.

Since CIDOC-CRM does not fully suffice for the purposes of TRANSRAZ (REQ1 and REQ2), the model has been extended by adding further properties and classes. The class *transraz:Event* is defined as subclass of *crm:E2 Temporal Entity* to represent cultural, political, military and further events that can be associated with points in time. *transraz:Event* inherits all properties of *crm:E2 Temporal Entity* to connect to other happenings, and additionally presents properties to connect events with agents, e.g. *transraz:participant* (REQ1). Moreover, the class *crm: E5 Event* is extended with two additional subclasses. The class *transraz:Relabeling* comprises changes of names of the instances of agents and objects. And the class *transraz:Refunction* addresses the changes on the function level, e.g. a building changes it function from pharmacy to shop (REQ2).

In order to address the uncertainty and incompleteness of time units in historical data, time expressions of different levels of granularity have to be considered. Thus, the new property *transraz:time* connects events with *time:TemporalEntity* from OWL-Time [16]. In contrast to *crm:E52 Time-Span*, it provides a more granular representation of time indications, it presents classes and relations for time instants and intervals, together with information about durations, and temporal position including date-time information. Additionally, the uncertainty and incompleteness of time intervals is addressed via providing temporal relations of the types interval-interval and interval-instant. Thus, the temporal relations can be modeled even if the exact event time is unknown, e.g. *interval1 time:overlaps interval2* (REQ4 and REQ5).

3.2.4. Representation and Annotation of Documents.

In the TRANSRAZ data model, the use of documents is two-fold:

- REQ1: Historical documents are themselves considered cultural heritage objects and require a semantic representation to enable the findability and interconnection of resources (CQ6.1, CQ6.3).
- REQ2: The provenience is required to keep track of the origin of statements and entities, the sources, they were derived from, are used as contextual evidence (CQ6.1-4).

Following the best practices, historical documents and their hierarchy are modeled with the FRBR-Aligned Bibliographic Ontology (FaBiO) [23]. FaBio provides rich semantics for recording and publishing bibliographic records. It enables the representation of archival records (*fabio:ArchivalDocumentSet* and *fabio:ArchivalDocument*), journals (*fabio:Journal*), their issues (fabio:JournalIssue) and articles (*JournalArticle*), books *fabio:Book* and their chapters (*fabio:BookChapter*). Partly making use of the DCMI Metadata Terms vocabulary[21] historical documents are annotated with the relevant metadata, e.g. *dcterms:creator, dcterms:publisher*. The Web Annotation Ontology[22] (OA) is used for providing annotations to the resources. Annotations (*oa:Annotation*) connect information to the source that the information was extracted from, for example a person mentioned in an address book or a fact derived from the NKL. Moreover, the property *oa:hasSelector* allows for linking an annotation to a specific part of a document including the text coordinates the annotation was made in. In this way, the provenance of a data excerpt can always be retrieved and proven. The task of representing knowledge from the rich, complex and heterogeneous resources, and, furthermore, enabling the usability and exploration of the information, presents diverse modeling challenges. To address them, in

[21] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/
[22] https://www.w3.org/ns/oa

the TRANSRAZ data model a modular modeling approach is developed and presented, that enables to benefit from the connection of data coming from different domains via domain-specific ontologies.

3.3. Evaluation of the TRANSRAZ Data Model

The competency questions introduced in Section 4.2 were developed to scope the data model, guide the modeling process, and, provide a way to evaluate the data model. The CQs were iteratively developed together with the domain experts – art historians and experts in digital art history. An evaluation of the data model is performed by examining if the CQs could be transformed into SPARQL queries and by verifying the appropriateness of the data model in delivering correct answers to the CQs[23][24]. The evaluation shows that the integration of the four main building blocks of the TRANSRAZ data model enables multifaceted queries. The use cases furthermore show that an interconnection between the building blocks is necessary and enables the most valuable findings. The presented contribution will not remain static and will be further enhanced and extended upon the integration of further resources in collaboration with domain experts.

4. Design Methodology and Use Cases

This section presents the methodology of data model development, discusses data sources and knowledge that serve as a basis of the data model and provides user-oriented interrogatives developed in the first stage of the the data model design.

4.1. Data and Data Sources as Basis of the TRANSRAZ Data Model

The TRANSRAZ data model aims at providing a semantic representation of the knowledge stored in historical resources, and, thus, is developed following a bottom-up approach. In this section, the selection of the currently processed data sources is introduced. The extraction and connection of the knowledge contained in these sources is analyzed and used to obtain modeling requirements of the data model.

TOPORAZ relational database is an essential source of the TRANSRAZ data model presented in this work. The database was developed as an effort of the predecending TOPORAZ project[25] and is hosted by FIZ Karlsruhe. Based on historical resources, such as city plans, cadastral plans, photographs, maps etc., domain experts collected and provided a small biography for every building in Nuremberg's main market square for four different time periods – 1620, 1811, 1910, and 2016. The data was then structured and manually inserted into subject-based tables. In particular, the database contains entities for architectural objects like streets and buildings, their residents and owners, constructing events (e.g. destruction and creation) and connecting time identification (e.g., "before 1890"), etc. in form of around 50 classes. In the ongoing TRANSRAZ project the area of the coverage has been extended to the reconstruction of 3000 buildings, which is 30 times more than the area of TOPORAZ. In contrast to TOPORAZ, contextual infor-

[23] CQs and SPARQL queries: `https://ise-fizkarlsruhe.github.io/Transraz/usecases`
[24] The SPARQL Endpoint: `https://www.toporaz.de/sparql`
[25] `https://www.toporaz.de/toporaz-current/`

mation about Nuremberg's inhabitants, events and buildings are extracted and integrated into the TRANSRAZ VRE automatically. Furthermore, more diverse data sources are analyzed and connected, which increases the heterogeneity and complexity of the task.

Address Books of Nuremberg is one of the first printed sources on the residents of Nuremberg starting with 1792. The address books are physically stored in the Nuremberg City Archives[26] and in The Germanisches Nationalmuseum[27], and are also provided digitally in a scanned image form. The books contain important information about persons, e.g. their addresses, professions, places of ownership and work, etc., and about companies, e.g. their names, industries, addresses, etc. Due to the complicated nature of the books, such as bad paper quality, distortion of pages, poor inking, as well as challenging linguistic features: Gothic fonts, ligatures, archaic terms, old spelling variants, abbreviations and typos, the extraction and structuring of the knowledge hidden in the books is a challenging task.

Nuremberg Artists Lexicon. The "Nürnberger Künstlerlexicon" (NKL) [13] is a collection of bibliographical articles about artists of Nuremberg based on various archival records ranging from the 12th century to the mid 20th century. The articles provide both personal information of artists such as addresses, professions, birth and death places and dates, family relations, places and periods of study, and information about their artworks and their public life. The articles of NKL are based on administrative records, the text is saturated with temporal units to describe the events, e.g., date of marriages and artworks creation, periods of study and work, change in the ownership of properties.

Journal of the Association for History of the City of Nuremberg. The "Mitteilungen des Vereins für Geschichte der Stadt Nürnberg" (MVGN)[28] is a journal focused on the history of Nuremberg. It has been publishing scholarly articles since 1879 and includes an annual issue with up to 40 reviews on significant historical events, prominent individuals and organizations of Nuremberg, and important buildings like St. Sebaldus Church. The articles are characterized by their complex sentence structures, filled with coordinating conjunctions and descriptive phrases.

Books of Nuremberg's Twelve Brothers. The "Nürnberg Zwölfbruderbücher"[29] are medieval books that feature portraits and biographical information about retired Nuremberg craftsmen residing in an old people's home. These books have been digitized, transcribed, and indexed[30]. They provide details such as the craftsmen's names, professions, birth and death dates and places, registration dates in the retirement home, and length of stay. Descriptions of the individuals' portraits are also included.

Nuremberg Old Town Reports. The "Nürnberger Altstadtberrichte" are reports that were first published in 1976 by the Altstadtfreunde Nürnberg Association[31]. They document significant construction projects through images, photographs, and text, while also featuring essays on the history, art, and culture of Nuremberg's old town. The reports extensively describe historical buildings, including their addresses, functions, and construction events, but also cover historical events, festivals, and individuals or companies associated with the architectural objects.

[26] https://www.nuernberg.de/internet/stadtarchiv_e/
[27] https://www.gnm.de/en/museum/
[28] https://www.bayerische-landesbibliothek-online.de/mvgn
[29] https://hausbuecher.nuernberg.de/index.php?do=page&mo=2
[30] https://hausbuecher.nuernberg.de/index.php?do=page&mo=5
[31] https://www.altstadtfreunde-nuernberg.de/de/home

4.2. Competency Questions of the TRANSRAZ Data Model

The development of the TRANSRAZ data model followed a user-centered design and evaluation methodology ([3]). Users of the data model are on the one hand the general public, generally interested in he city's history, architecture and its inhabitants. On the other hand, users are researchers in the fields of history, social science, digital humanities, and architecture. With the goal to scope the desired data model and provide future users with the ability to gain answers through exploring the data model and its associated knowledge, a set of competency questions (CQs) in collaboration with domain experts was developed incrementally and iteratively, and resulted in 6 categories:

1. **Ancestor Search.** Users with ancestors in Nuremberg have the possibility to explore, where their relatives lived, how they lived and what their neighborhood looked like.

   CQ1.1: What information is available for a specific person or family living in the city? This involves events related to the person, family relations, their occupation.
   CQ1.2: Who lived in a certain house or at a certain address?
   CQ1.3: What did a certain neighborhood look like at a certain year? What kind of industries were based there?

2. **Wealth Distribution.** Researchers who are interested in the distribution of wealth throughout the city can explore the location of industries, infrastructure and living areas of Nuremberg's inhabitants.

   CQ2.1: Which properties and real estates belonged to a person or family and how did they distribute throughout the city?
   CQ2.2: Who owned properties and real estates in a certain street or district?
   CQ2.3: Where did people of certain occupations (e.g. bakers, teachers) live?
   CQ2.4: Where did specific industries and branches (e.g. schools, breweries) establish?

3. **Change and Events.** Researchers interested in the transformation of the city and changes of entities can explore the movement of a person within the city over the years, the destruction and construction of architectural objects and social events taking place.

   CQ3.1: Where did one person live in the city throughout time?
   CQ3.2: Where were specific infrastructures like hospitals and schools located in a certain time? And how their establishment developed throughout time?

4. **Architecture and Infrastructure.** Users intending to learn about the city architecture and infrastructure development can query architectural objects and the development ob buildings and their functions over time.

   CQ4.1: In what area of the city were the highest buildings (by number of of floors)?
   CQ4.2: What different roof types were built throughout the years in the city?
   CQ4.3: What building function did a building have over the years (e.g. pharmacy, hotel) and how was this documented in the historical sources?

5. **Social and Cultural.** To learn about the social life of the historical city over the years, users are able to explore touristic and culturally important areas.

   CQ5.1: What were the main cultural areas in the city, e.g. cultural and art institutions?

   CQ5.2: Which parts of the city did most artists live in?

6. **Historical Sources and Provenance.** Researchers can verify findings presented in the VRE by directly exploring the connected historical source documents. Furthermore, document annotations and their origins can be reproduced.

   CQ6.1: What historical literature is available for a certain street, building. person?

   CQ6.2: Where can information about a certain street, building, person be found in the historical resource, e.g. page number or coordinates?

   CQ6.3: What types of documents are available that mention a certain concept, e.g. certificates, archival records?

   CQ6.4: How and when were the annotations of a certain concept created? Manually or automatically?

5. Conclusion and Future Work

Within the TRANSRAZ project the TRANSRAZ data model has been created, which enables semantic representation of agents, events, architectural objects, and historical documents of historical cities. This model enables scientifically accurate exploration of the Nuremberg's history in a 3D virtual environment, and can be applied to other historical cities with similar aims. It is available publicly, it is planned to extend it by adding new vocabularies and controlled vocabularies for consistency in historical terms. Moreover, the TRANSRAZ Knowledge Graph integrated into TOPORAZ VRE will be extended with addition of further data resources. Exploring the knowledge extracted from the historical resources and their interconnection with a 3D VRE will allow users to obtain an impression of the hidden past and, hence, better understand the formation of today's world, its society and oneself.

References

[1] Auer, S., Lehmann, J., Hellmann, S.: Linkedgeodata: Adding a spatial dimension to the web of data. In: Int. Semantic Web Conference. pp. 731–746. Springer (2009)

[2] Behley, J., Steinhage, V.: Generation of 3D city models using domain-specific information fusion. In: Int. Conference on Computer Vision Systems. pp. 164–173. Springer (2009)

[3] Bravo, M., Luis, H.R., Reyes, J.: Methodology for ontology design and construction. Contaduría y Administración **64**, 134 (03 2019).

[4] Bruns, O., Tietz, T., Chaabane, M.B., Portz, M., Xiong, F., Sack, H.: The Nuremberg Address Knowledge Graph. In: 18th Extended Semantic Web Conference (ESWC), Poster and Demo Track. Springer (2021)

[5] Bruns, O., Tietz, T., Vafaie, M., Dessí, D., Sack, H.: Towards a representation of temporal data in archival records: Use cases and requirements. In: Proc. of the Int. Workshop on Archives and Linked Data. pp. 128–134 (2021)

[6] Carriero, V.A., Gangemi, A., Mancinelli, M.L., Marinucci, L., Nuzzolese, A.G., Presutti, V., Veninata, C.: ArCo: The Italian cultural heritage knowledge graph. In: Int. Semantic Web Conference. pp. 36–52. Springer (2019)

[7] Diara, F., Rinaudo, F.: Open Source HBIM for Cultural Heritage: a project proposal. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences **42**(2) (2018)

[8] Doerr, M.: The CIDOC CRM, an ontological approach to schema heterogeneity. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2005)

[9] Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C., Van de Sompel, H.: The europeana data model (edm). In: World Library and Information Congress: 76th IFLA general conference and assembly. vol. 10, p. 15 (2010)

[10] Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S., Demidova, E.: WorldKG: A World-Scale Geographic Knowledge Graph. In: Proc. of the 30th ACM Int. Conference on Information & Knowledge Management, pages=4475–4484, year=2021

[11] Gottschalk, S., Demidova, E.: EventKG: a multilingual event-centric temporal knowledge graph. In: European Semantic Web Conference. pp. 272–287. Springer (2018)

[12] Graves, M., Constabaris, A., Brickley, D.: FOAF: Connecting people on the semantic web. Cataloging & classification quarterly **43**(3-4), 191–202 (2007)

[13] Grieb, M.H.: Nürnberger Künstlerlexikon: Bildende Künstler, Kunsthandwerker, Gelehrte, Sammler, Kulturschaffende und Mäzene vom 12. bis zur Mitte des 20. Jahrhunderts. Walter de Gruyter (2011)

[14] Gutierrez, C., Hurtado, C., Vaisman, A.: Temporal rdf. In: European Semantic Web Conference. pp. 93–107. Springer (2005)

[15] Hartig, O.: Foundations of RDF* and SPARQL*:(An alternative approach to statement-level metadata in RDF). In: AMW 2017 11th Alberto Mendelzon Int. Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. vol. 1912. Juan Reutter, Divesh Srivastava (2017)

[16] Hobbs, J.R., Pan, F.: Time ontology in OWL. W3C working draft **27**(133), 3–36 (2006)

[17] Honneth, A.: Reification: A new look at an old idea. Oxford University Press (2008)

[18] Hyvönen, E., Mäkelä, E., Kauppinen, T., Alm, O., Kurki, J., Ruotsalo, T., Seppälä, K., Takala, J., Puputti, K., Kuittinen, H., et al.: CultureSampo—Finnish culture on the Semantic Web 2.0. Thematic perspectives for the end-user. In: Proceedings, Museums and the Web. pp. 15–18 (2009)

[19] Iannella, R., McKinney, J.: vCard Ontology-for describing People and Organizations. W3C Group Note NOTE-vcard-rdf-20140522 (2014)

[20] Karalis, N., Mandilaras, G., Koubarakis, M.: Extending the YAGO2 knowledge graph with precise geospatial knowledge. In: Int. Semantic Web Conference. pp. 181–197. Springer (2019)

[21] Kim, S., Ahn, J., Suh, J., Kim, H., Kim, J.: Towards a semantic data infrastructure for heterogeneous Cultural Heritage data-challenges of Korean Cultural Heritage Data Model (KCHDM). In: 2015 Digital Heritage. vol. 2, pp. 275–282. IEEE (2015)

[22] Kröber, C., Messemer, H., Bruschke, J., Maiwald, F., Niebling, F., Münster, S.: Ein 4D-Browser für historische Photographien: NutzerInnen zwischen medialer Vermittlung und Forschung, pp. S. 25–27. arthistoricum.net, Heidelberg (2020).

[23] Peroni, S., Shotton, D.: FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. Journal of Web Semantics **17**, 33–43 (12 2012)

[24] Razum, M., Göller, S., Sack, H., Tietz, T., Vsesviatska, O., Weilandt, G., Grellert, M., Scharm, T.: TOPORAZ: Ein digitales Raum-Zeit-Modell für vernetzte Forschung am Beispiel Nürnberg. Information-Wissenschaft & Praxis **71**(4), 185–194 (2020)

[25] Tan, M.A., Tietz, T., Bruns, O., Oppenlaender, J., Dessí, D., Sack, H.: DDB-KG: The German Bibliographic Heritage in a Knowledge Graph. In: The 6th Int. Workshop on Computational History (HistoInformatics2021-JCDL2021). Ed.: Y. Sumikawa (2021)

[26] Tietz, T., Bruns, O., Göller, S., Razum, M., Dessì, D., Sack, H.: Knowledge graph enabled curation and exploration of Nuremberg's city heritage. In: Proc. of the Conference on Digital Curation Technologies (Qurator 2021) (2021)

[27] Udrea, O., Recupero, D.R., Subrahmanian, V.: Annotated rdf. ACM Transactions on Computational Logic (TOCL) **11**(2), 1–41 (2010)

[28] Van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., Schreiber, G.: Design and use of the Simple Event Model (SEM). Journal of Web Semantics **9**(2), 128–136 (2011)

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230013

COBALT: A Content-Based Similarity Approach for Link Discovery over Geospatial Knowledge Graphs

Alexander BECKER [a], Abdullah AHMED [a] and Mohamed Ahmed SHERIF [a]
Axel-Cyrille NGONGA NGOMO [a]

[a] *DICE group, Department of Computer Science, Paderborn University*
ORCiD ID: Alexander Becker https://orcid.org/0009-0003-8212-4647, Abdullah Ahmed https://orcid.org/0000-0002-0668-0735, Mohamed Ahmed Sherif https://orcid.org/0000-0002-9927-2203, Axel-Cyrille Ngonga Ngomo https://orcid.org/0000-0001-7112-3516

Abstract. *Purpose:* Data integration and applications across knowledge graphs (KGs) rely heavily on the discovery of links between resources within these KGs. Geospatial link discovery algorithms have to deal with millions of point sets containing billions of points. *Methodology:* To speed up the discovery of geospatial links, we propose COBALT. COBALT combines the content measures with R-tree indexing. The content measures are based on the area, diagonal and distance of the minimum bounding boxes of the polygons which speeds up the process but is not perfectly accurate. We thus propose two polygon splitting approaches for improving the accuracy of COBALT. *Findings:* Our experiments on real-world datasets show that COBALT is able to speed up the topological relation discovery over geospatial KGs by up to 1.47×10^4 times over state-of-the-art linking algorithms while maintaining an F-Measure between 0.7 and 0.9 depending on the relation. Furthermore, we were able to achieve an F-Measure of up to 0.99 by applying our polygon splitting approaches before applying the content measures. *Value:* The process of discovering links between geospatial resources can be significantly faster by sacrificing the optimality of the results. This is especially important for real-time data-driven applications such as emergency response, location-based services and traffic management. In future work, additional measures, like the location of polygons or the name of the entity represented by the polygon, could be integrated to further improve the accuracy of the results.

Keywords. Knowledge graphs, Data Integration, Linked Data, Geospatial Knowledge graphs, Content Measure Similarity, Topological Relations

1. Introduction

The necessity for highly scalable methods for finding links between geospatial resources has arisen as a result of the rapid proliferation of linked geospatial data. Only 7.1% of the relationships between resources relate geographical elements, as was noted in earlier publications [1]. There are two basic causes for this: I) The vast quantity of geospatially represented resources on Linked Open Data (LOD) necessitates scalable techniques for computing linkages between geospatial resources. For example LINKEDGEODATA [2] has over 20 billion triples describing millions of geographical things. II) The computation of certain relations, such as distance and topological links between geospatial resources, have to deal with the vector representation of geospatial data. For instance, for identifying the nearby points of interest within a certain radius.

Discovering links among KGs in RDF is crucial for many data-driven applications, according to the Linked Data principles [3]. Nowadays dealing with geospatial resources is a fundamental in many real-time applications [4], such as emergency response, location-based services and real-time traffic management. However, generating links among such real-time geospatial KGs is challenging task in order to enable real-time decision making. Thus, both the efficiency and scalability of the link discovery process becomes more challenging.

Recently, algorithms such as RADON [5], RADON2 [6], GIA.NT [7], and DORIC [8] have been developed. These algorithms compute topological relations between geographical resources quickly and effectively. In all of them, the *Dimensionally Extended Nine-Intersection Model* (DE-9IM) [9] is used. The DE-9IM defines topological relations between two-dimensional geometries by calculating the dimensions of the intersections between the interior, boundary and exterior of two geometries. The relations defined by the DE-9IM are the ones commonly used in natural language [10]: `Equals`, `Disjoint`, `Intersects`, `Touches`, `Crosses`, `Within`, `Contains`, `Overlaps`, `Covers` and `Covered By`. In an attempt to speedup the computation of geospatial relations, Ahmed et al. [11] have studied the effect of simplifying the resources' geometries on the runtime and F-Measure of link discovery approaches. However, computing the DE-9IM is still very expensive in terms of runtime.

In this paper, we propose COBALT, an approach based on the content measures combined with R-tree indexing to discover the topological relations defined in [9] and [10]. To the best of our knowledge, this is the first work that uses content-based measures integrated with R-tree indexing for discovering links among RDF geospatial resources. We summarize our contribution as follows:

1. We present and formalize the problem of topological relation discovery for geospatial resources based on content measures and R-tree indexing.
2. We study the effect of using different R-tree building algorithms, node capacities and the impact of indexing both datasets.
3. We study the impact of using the content-based measures for topological relations discovery on both runtime and accuracy.
4. In order to increase the accuracy of our approach, we propose two polygon splitting strategies and analyze the effect of them on both runtime and accuracy.

The rest of the paper is structured as follows. We begin by introducing the link discovery problem over RDF KG in Section 2, where we also formally define the topological relations based on content measures. Then, we describe our approach in Section 3. In Section 4, we present our evaluation and results. We then discuss the state-of-the-art related work in Section 5. Finally, we conclude our paper and present some future work in Section 6. Our implementation of COBALT is open source and implemented into the LIMES framework.[1]

[1] `https://github.com/dice-group/LIMES`

Figure 1. The content measure relations.

2. Preliminaries

Knowledge Graph. A Knowledge Graph (KG) G is a set of triples $(s,p,o) \in (\mathscr{R} \cup \mathscr{B}) \times \mathscr{P} \times (\mathscr{R} \cup \mathscr{L} \cup \mathscr{B})$, where $\mathscr{R}$ is the set of all resources, $\mathscr{B}$ is the set of all blank nodes, $\mathscr{P}$ is the set of all predicates, and $\mathscr{L}$ the set of all literals.

Link Discovery. Given a source knowledge graph G_s and a target knowledge graph G_t (for example, two KGs containing the geometric representation of national borders) and a relation r (e.g., `:touches`), the goal of the link discovery problem is to find all pairs $(s,t) \in G_s \times G_t$ such that $r(s,t)$ holds. The result is produced as a set of links called a *mapping*: $M = \{(s,r,t) | s \in G_s, t \in G_t\}$.

Content Measures for Topological Relations. We use the content measures as defined Godoy et al. [12] for deciding if the relation $r(s,t)$ exists. Godoy et al. have implemented three content measures to determine whether a topological relation between two polygons exists by comparing the area, the diagonal or the area and the diagonal of the polygon's minimum bounding boxes. The relations distinguished by the content measures are shown in Figure 1. *The Minimum Bounding Box (MBB)* of a polygon P with $n \geq 3$ points $((x_1,y_1),\ldots,(x_n,y_n))$ is defined as the smallest rectangle which fully contains the polygon's points. Formally, $\mathrm{MBB}(P) = \big((X_{min},Y_{max}),(X_{max},Y_{min})\big)$, where $X_{min} = \min(x_1,\ldots,x_n)$, $X_{max} = \max(x_1,\ldots,x_n)$, $Y_{min} = \min(y_1,\ldots,y_n)$ and $Y_{max} = \max(y_1,\ldots,y_n)$. The *area* of a MBB M is defined as $area(M) = (X_{max} - X_{min}) \cdot (Y_{max} - Y_{min})$. The *diagonal* of a MBB M is formally defined as $diagonal(M) = \sqrt{(X_{max} - X_{min})^2 + (Y_{max} - Y_{min})^2}$. We further use the same definitions of *intersection*, *union* and *distance* among MBBs from [12].

3. Approach

We start our approach by indexing the source dataset polygons using R-tree then we apply content measure on the indexed polygons.

R-tree Indexing. R-trees [13] are an enhanced variant of binary trees, where an R-tree stores the MBBs of the polygons instead of the polygons themselves. In COBALT, we use *Guttman*'s R-tree [14] to index the source dataset in order to filter out as many disconnected polygon pairs as possible to reduce the runtime of the linking process. Every node's MBB contains all its children's MBBs, so in case an MBB of a parent node does not intersect a query rectangle (a query rectangle is a MBB from target data), none of its descendants can [14]. The bottom layer of an R-tree stores the MBBs of source dataset polygons, and all layers above it match the criterion applied to indexing the bottom layer. One example of a handcrafted R-tree is depicted in Figure 2.

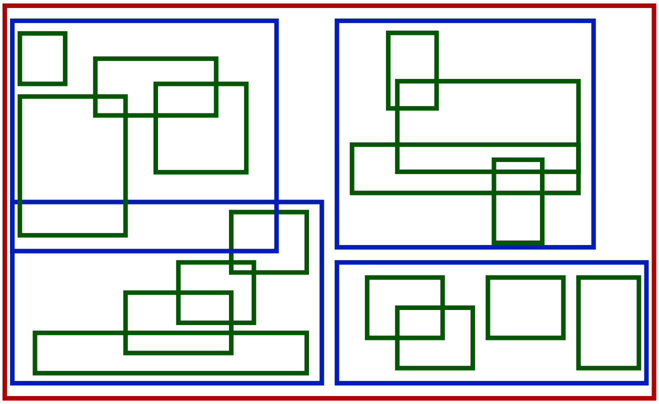


Figure 2. A handcrafted R-tree, green is the bottom layer, blue is the middle, and red is the top layer.

Querying R-tree. R-trees are easy to query recursively. Let q be the target MBB of the polygon for which we want to find the intersected MBBs of the source polygons. Since the nodes of an R-tree nodes are R-trees, we utilize the same algorithm for each node. If the current node of the R-tree is at the bottom layer, (i.e., it contains no other R-trees but polygons), we then verify if q intersects each of the MBBs of the source polygons saved in this node and add such source polygons to the query result. In case the current node of the R-tree is not on the bottom layer, we check if each child node's MBB has at least one common point with q and if that is the case, we recursively repeat the method for that node. In Figure 2 for instance, the MBBs of the left two blue nodes overlap. In case the query rectangle q lies in the area where two nodes' MBBs overlap, we have to check the children of both nodes. Therefore, we need a fast building approach that minimizes overlapping parent nodes.

Building R-tree. There are two main ways for constructing R-trees: (i) *Static building algorithms* work by getting all data as the input and then constructing the tree with all data at once. (ii) *Dynamic building algorithms* work by inserting data one by one into the tree. As our datasets are not changing frequently, we focus on static algorithms as dynamic algorithms will require more run time for reinserting data to keep the R-tree balanced. Because we only query the R-tree once for every target geometry, the build quality (overlap) is not as important for an overall fast execution. To test the impact of different building algorithms, we use four static R-tree building algorithms (i.e., *SmallestX*, *STR*, *OTM* and *PackedHilbertR-tree*) and one dynamic algorithm (i.e., R*-Tree). The first static algorithm is *SmallestX* [15], which sorts the MBBs by the smallest x coordinate. The *SortTileRecursive* [16] algorithm (STR) builds the R-tree bottom-up and divides the MBBs into slices sorted by the x coordinate and then sorts them by the y coordinate, then recursively combines the parent nodes of the bottom layer. The *OTM* algorithm [17] works similar to STR, but recursively sorts the MBBs by alternating x and y coordinate with a top-down bulk loading approach. The last static building algorithm we use is the *PackedHilbertR-tree* [18] algorithm, which sorts the MBBs by their position on the Hilbert curve. On the other hand, we use the dynamic R-tree building algorithm *R*-Tree* [19], which supports inserting of new elements after creation and tries to minimize the area occupied by nodes. In our experiments, we insert the polygons one

Relation	Disjoint	Meets	Overlap	Equals	Covers	CoveredBy	Contains	Inside
$F_a(A,B)$	(0,1)	(0,1)	(0,1)	1	1	(0,1)	1	(0,1)
$F_a(B,A)$	(0,1)	(0,1)	(0,1)	1	(0,1)	(0,1)	(0,1)	1
$F_a(A,B)+F_a(B,A)$	(0,1)	(0,1]	(0,2)	2	(1,2)	(1,2)	(1,2)	(1,2)

Table 1. Area based content measure relations based on values of F_a. [12]

by one into the R*-Tree and use the values $\{4, 8, 16, 32, 64, 128, 256\}$ for the capacities of each node.

Content Measures For Topological Relations. Given two MBBs A and B, the *area-based content measure* (F_a) is the first of the three content measures from [12]. F_a is the normalization of the area of each MBBs of both A and B by the area of the MBB of the union of A and B. Formally,

$$F_a(A,B) = \frac{\text{area(A)}}{\text{area(MBB(A} \cup \text{B))}}, \tag{1}$$

where $F_a(B,A)$ is defined analogously. The range of $F_a \in (0,1]$. In Table 1, we present the values of $F_a(A,B)$, $F_a(B,A)$ and $F_a(A,B)+F_a(B,A)$ for the different topological relations. F_a cannot distinguish the following pairs of relations: `(Meet, disjoint)`, `(covers, contains)` and `(covered by, inside)`. For instance, the union MBB will be the same as the MBB of the MBBs for `contains` and `covers`. However, this measure can accurately detect the `equal` relation because both the input MBBs have the same area as their union's MBB. The second content measure is the *diagonal-based content measure* (F_d) [12], formally defined as:

$$F_d(A,B) = \frac{\text{diagonal(A)}}{\text{diagonal(MBB(A} \cup \text{B))}}. \tag{2}$$

The range of $F_d \in (0,1]$ and it cannot distinguish `(covers, contains)` and `(covered by, inside)` for the same reason as in the case of F_a. The third content measure is the *mixed content measure* (F_m) [12]. F_m utilizes the area, diagonal and distance of the MBBs for finding the topological relations. Unlike the other two content measures, it is able to distinguish between `(contains, covers)` and `(inside, covered by)`. Formally,

$$F_m(A,B) = \frac{\text{area(A)} - 2 \cdot \text{area(MBB(A} \cap \text{B))}}{\text{area(A)}} + \frac{\text{distance(A, B)}}{\text{diagonal(A)}}. \tag{3}$$

Combining R-tree Indexing and Content Measure. Our R-tree indexing filters out `disjoint` polygon pairs based on their MBB. We only keep the indexed source dataset in memory, which reduces the space complexity as we then stream-process the target dataset. In the case of the `disjoint` relation, we first add all pairs of geometries that the indexing would filter out to the result set then we check the other relations on the rest of the geometries pairs. In Algorithm 1, we line out the steps for the area based content measure F_a. For the other measures, we replace F_a with F_d for the diagonal measure and F_m for the mixed measure (Lines 11-12). Additionally, the values of F_a need to be checked against the other measures' values from [12] (Line 14).

Algorithm 1 DiscoverLinksAreaBased(G_s, G_t, r)

Input: Source KG G_s, Target KG G_t, Topological relation r
Output: Mapping : $M = \{(s,r,t) | s \in G_s, t \in G_t\}$
$tree \leftarrow buildRtree(G_s)$
Initialise $M \leftarrow \{\}$
for each MBB(t) $t \in G_t$ **do**
  $I \leftarrow$ queryRtree(tree, t))
  if r is disjoint **then**
    Add all pairs$(s,r,t) \forall s \in (G_s \setminus I)$ to M
  end if
  for each MBB(s) $s \in I$ **do**
    $X \leftarrow \mathrm{F}_a(MBB(s), MBB(t))$
    $Y \leftarrow \mathrm{F}_a(MBB(t), MBB(s))$
    $Z \leftarrow X + Y$
    if X, Y, Z match the respective values of the relation r in in Table 1 **then**
      Add (s,r,t) to M
    end if
  end for
end for
return M

Algorithm 2 MatchTrees(sourceTree, targetTree)

Result $\leftarrow \{\}$
if area(sourceTree) < area(targetTree) **then**
  swap sourceTree and targetTree
end if
for each child of sourceTree **do**
  if child is leaf **then**
    Result = Result $\cup$ queryRtree(targetTree, MBB(child))
  else
    Result = Result $\cup$ MatchTrees(child, targetTree)
  end if
end for
return Result

Indexing both Datasets. In many cases swapping the source and target datasets results in different runtimes. In order to reduce the impact of dataset ordering on the runtime, we study the possibility of indexing both datasets instead of one. Instead of querying the R-tree for each target geometry, we use Algorithm 2 to match two R-trees to each other and recursively find all pairs that intersect. This approach removes the need to choose which dataset to index but comes with the price of increasing the memory footprint of our approach as we have to keep both datasets in memory.

Splitting Polygons to Gain Accuracy. We are able to improve the F-Measure of Cobalt for some relations by splitting the geometries into multiple pieces before using the content measure functions to determine the relation. In particular, we split polygons recursively t times into four pieces using two different strategies: **1) Equal split**, where

(a) Splitting the polygon into equal size parts two times

(b) Splitting the polygon into fitting parts two times

Figure 3. Different options to split polygons. The green triangle is the original polygon, the dark red lines are the splitting lines for the first split iteration, the light red lines are the splitting lines for the second split iteration. The blue rectangles indicate the MBB used for determining the second iteration splitting lines of the top left corner.

we are splitting the original polygon into equally sized parts. The resulting polygon parts are the intersection of a grid pattern over the original polygon and the original polygon itself. In some cases, this leads to some splits not achieving any additional information as their parts of the grid are empty. For instance in Figure 3a, any further splits of the top left corner cell (the blue highlighted cell) would not increase the accuracy of COBALT as the original polygon does not have any points within this cell. **2) Fitting split**, where we divide the polygon into equally sized parts but using the MBB of the current polygon part for further splitting. Splitting the top left corner cell of the same polygon of the previous example using this strategy will result in the splitting presented in Figure 3b, where further splitting of the blue highlighted cell results in more detailed splits that fit better to the shape of the polygon. After splitting the polygons we compute the MBBs for all parts. Now as we have multiple polygon parts, we change the way the relation of the original polygon is determined. Let t be the number of splits into four parts. Let $A_{(i,j)}$ be the split part of geometry A at column i and row j and $B_{(k,l)}$ be the split part of geometry B at column k and row l for $\{(i,j,k,l) \in \mathbb{N}^4 | 0 \leq i,j,k,l < 2^t\}$. The newly defined relations can be found in Table 2. In particular, every grid pattern $A_{(i,j)}$ must be equal to $B_{(k,l)}$ for the `equals` relation to hold. For the `intersects` relation, at least one $A_{(i,j)}$ has to intersect with at least one $B_{(k,l)}$. For the `within` relation, all $A_{(i,j)}$ have to be contained in the MBB of the union of all $B_{(k,l)}$ it intersects. For the `contains` relation, we swap A and B then compute `within` relation instead. For the `overlaps` relation, we three conditions must hold: 1) at least one $A_{(i,j)}$ is not within B, 2) at least one $B_{(k,l)}$ is not within A, and 3) at least one $A_{(i,j)}$ intersects with at least one $B_{(k,l)}$. For the *touches* relation, at least one $A_{(i,j)}$ must touch any $B_{(k,l)}$ and every $B_{(i,j)}$ is related to every $B_{(k,l)}$ by either `touches` or `disjoint` relation.

4. Evaluation & Results

Datasets. We use two real-world datasets for evaluating COBALT: 1) The *NUTS*[2] dataset from the *Eurostat* group describes the territory of countries in the European Union, (potential) candidate countries and countries belonging to the European Free

[2] https://ec.europa.eu/eurostat/de/web/gisco/geodata/reference-data/administrative-units-statistical-units/NUTS, accessed on 01.09.2022

`equals`	$\forall\, \{(i,j) \in \mathbb{N}^2 \mid 0 \leq i,j < 2^t\}$: $A_{(i,j)}$ `equals` $B_{(i,j)}$
`intersects`	$\exists\, \{(i,j,k,l) \in \mathbb{N}^4 \mid 0 \leq i,j,k,l < 2^t\}$: $A_{(i,j)}$ `intersects` $B_{(k,l)}$
`disjoint`	$\forall\, \{(i,j,k,l) \in \mathbb{N}^4 \mid 0 \leq i,j,k,l < 2^t\}$: $A_{(i,j)}$ `disjoint` $B_{(k,l)}$
`within`	$\forall\, \{(i,j) \in \mathbb{N}^2 \mid 0 \leq i,j < 2^t\}$: $A_{(i,j)}$ `within` $\mathrm{MBB}(\{B_{(k,l)} \mid \forall \{(k,l) \in \mathbb{N}^2 \mid 0 \leq k,l < 2^t\}:$ $A_{(i,j)}$ `intersects` $B_{(k,l)}\})$
`contains`	swap A and B then compute `within`
`overlaps`	$(\exists\, \{(i,j,k,l) \in \mathbb{N}^4 \mid 0 \leq i,j,k,l < 2^t\}$: $A_{(i,j)}$ `equals` $B_{(k,l)}) \vee A_{(i,j)}$ `within` $B_{(k,l)}) \vee$ $A_{(i,j)}$ `contains` $B_{(k,l)}) \vee A_{(i,j)}$ `overlaps` $B_{(k,l)}) \wedge$ $(\exists\, \{(i,j) \in \mathbb{N}^2 \mid 0 \leq i,j < 2^t\}$: $A_{(i,j)}\ \neg$ `within` $\mathrm{MBB}\ (\{B_{(k,l)} \mid \forall \{(k,l) \in \mathbb{N}^2 \mid 0 \leq k,l < 2^t\}:$ $A_{(i,j)}$ `intersects` $B_{(k,l)}\}))\wedge$ $(\exists\, \{(i,j) \in \mathbb{N}^2 \mid 0 \leq i,j < 2^t\}$: $B_{(i,j)}\ \neg$`within` $\mathrm{MBB}(\{A_{(k,l)} \mid \forall \{(k,l) \in \mathbb{N}^2 \mid 0 \leq k,l < 2^t\}:$ $B_{(i,j)}$ `intersects` $A_{(k,l)}\}))$
`touches`	$(\exists\, \{(i,j,k,l) \in \mathbb{N}^4 \mid 0 \leq i,j,k,l < 2^t\}$: $A_{(i,j)}$ `touches` $B_{(k,l)})\wedge$ $\neg(\exists\, \{(i,j,k,l) \in \mathbb{N}^4 \mid 0 \leq i,j,k,l < 2^t\}$: $A_{(i,j)}$ `equals` $B_{(k,l)}) \vee A_{(i,j)}$ `within` $B_{(k,l)}) \vee$ $A_{(i,j)}$ `contains` $B_{(k,l)}) \vee A_{(i,j)}$ `overlaps` $B_{(k,l)})$

Table 2. Topological relations based on multiple splits of polygons.

Trade Association. 2) The *Corine Land Cover (CLC)*[3] [20] created by the *European Environment Agency*. CLC contains information about the land use of the 39 EEA39 countries[4].

Hardware & Software. All experiments were conducted on the NOCTUA1[5] cluster of the *Paderborn university*. NOCTUA1 consists of 256 compute nodes, each having two Intel *Xeon Gold* "Skylake" 6148 processors, which comes to a total of 40 cores with 2.4 GHz and 192 GiB main memory. All used algorithms were implemented in *Java*, and the compute nodes ran on *OpenJDK* version 11.0.2. For an accurate runtime measurement, all experiments were started with all datasets already loaded into the main memory. Additionally, for linking each dataset pair, we ran the algorithms on the same compute node. All experiments were conducted with a memory limit of 30 GB. Unless otherwise stated, we use only one core for all the experiments.

Experiments Settings. We use COBALT$_{area}$, COBALT$_{diagonal}$ and COBALT$_{mixed}$ to dub the *area*, *diagonal* and the *mixed* measures of COBALT, respectively. We use the following four baselines: i) RADON [5], ii) RADON with only the MBBs of the original polygons (dubbed RADON$_{\mathrm{MBB}}$), iii) GIA.NT [7] and iv) GIA.NT with only the MBBs of the original polygons (dubbed GIA.NT$_{\mathrm{MBB}}$). For a fair runtime comparison, we use a version of GIA.NT that computes only one relation at a time. We also implemented a space-indexing-based version of COBALT, where we optimized the content-based measures based on the space tiling indexing of RADON [5]. We use COBALT$_{area(R)}$, COBALT$_{diagonal(R)}$ and COBALT$_{mixed(R)}$ to dub the *area*, *diagonal* and the *mixed* measures of the space-tiling-based indexing measures of COBALT, respectively. We also used the *Douglas-Peucker* polygon simplification algorithm [21]. The simplification is ap-

[3]https://land.copernicus.eu/pan-european/corine-land-cover, accessed on 01.09.2022
[4]https://land.copernicus.eu/portal_vocabularies/geotags/eea39
[5]https://pc2.uni-paderborn.de/hpc-services/available-systems/noctua1

plied to the dataset using simplification thresholds {0.05, 0.1, 0.2}, then the relations are computed using RADON. They were labeled as $\text{RADON}_{\text{simp}(0.05)}$, $\text{RADON}_{\text{simp}(0.1)}$ and $\text{RADON}_{\text{simp}(0.2)}$. Within all experiments, we computed the topological relations `{equals, intersects, contains, within, touches, overlaps}`.

Research questions. We aim to answer the following research questions:

Q_1. What is the effect of indexing the input datasets on the runtime of COBALT?
Q_2. How much efficiency (i.e., less runtime) we gain by using content measure for topological relation discovery?
Q_3. How much accuracy we lose (i.e., less F-Measure) by using content measure for topological relation discovery?
Q_4. In case we use a simplified version of the original polygons, will we have a better trade-off between accuracy and efficiency than using COBALT?
Q_5. Will COBALT benefit from parallelization for big KGs such as CLC?
Q_6. What is the trade-off between accuracy and efficiency when we integrate our polygon splitting strategies into COBALT?

Research question Q_1. The aim of our *first set of experiments* was to evaluate different R-tree indexing options for COBALT. To measure the difference in runtime between indexing only one dataset vs. indexing both the source and target datasets, we linked NUTS to CLC (see Table 3) and CLC to NUTS (see Table 4). First, we compared the algorithms that index both datasets to the algorithms that only index one dataset. Our results showed that when linking NUTS to CLC most of the algorithms that only index one dataset are faster than the matching algorithms that index both of them. On the CLC to NUTS experiment (CLC×NUTS), however, the matching algorithms that index both datasets are faster than the algorithms that only index one dataset. This shows that the choice of the source dataset makes a difference regarding the runtime and the smaller dataset should be indexed instead of the bigger one. Because of the higher memory need for indexing both datasets, we decided to index only one dataset in our further experiments. In addition, our results showed that computing the *Hilbert curve*, or inserting entries one by one with the *R*-Tree*, takes much more time than the other algorithms. OMT and STR have the best runtime of the algorithms as their computations are not expensive and produce high quality R-trees. We conclude that the choice of the R-tree building algorithm as well as the capacity of the R-tree are highly dependent on the datasets used for benchmarking. It is important to find a balance between a fast building algorithm and an algorithm that allows efficient queries. Sorting entries by both x and y coordinate like OMT and STR is a good way to achieve this. This answers our first research question Q_1.

For the following experiments we use the STR building algorithm with a capacity of 4, but to respect the downside of only indexing one dataset, we also use the longer taking dataset combination for runtime values.

To answer Q_2, Q_3 and Q_4, we conducted our *second set of experiments* where we evaluate the performance of COBALT vs. all the baselines in terms of runtime and F-Measure. In particular, we aim to find the topological relations within the NUTS dataset against itself (i.e., NUTS×NUTS) and CLC×NUTS using each of the aforementioned algorithms. For linking NUTS×NUTS, the total required runtimes to compute the topological relations are shown in Figure 4a. The F-Measure for each relation can be seen in Table 5. For linking CLC×NUTS, the total required runtimes to compute the same six relations are shown in Figure 4b. The F-Measure of each relation can be seen in Table 6.

Table 3. Runtime in milliseconds for linking NUTS to CLC using different R-tree building algorithms and capacities combined with the mixed content measure

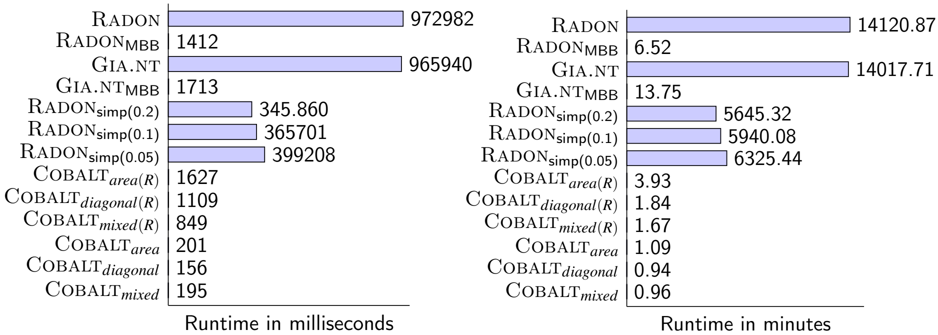
Algorithm	4	8	16	32	64	128	256
R*TREE	55745	49757	61810	77957	137733	202192	208227
HILBERT	258193	245062	233901	236357	363893	270579	273802
SMALLESTX	204226	95004	58875	64362	65360	62508	65948
OMT	35055	35778	36415	44063	71217	82937	103397
STR	36986	37088	38366	44042	48191	61899	77262
MATCHHILBERT	398502	345562	328975	320710	327772	357677	410224
MATCHSMALLESTX	85300	71626	62810	59780	46495	53344	66957
MATCHOMT	72359	56792	55213	50404	44134	48084	54525
MATCHSTR	45716	42130	41216	41769	41859	42278	43248

Table 4. Runtime in milliseconds for linking CLC to NUTS using different R-tree building algorithms and capacities combined with the mixed content measure

Algorithm	4	8	16	32	64	128	256
R*TREE	135037	139044	189439	372823	1067386	3527572	1067386
HILBERT	801618	866145	621211	793548	556130	540250	556130
SMALLESTX	263744	245692	422600	195218	93560	134091	324985
OMT	77928	62193	60354	55639	48598	54950	66138
STR	51448	47783	46708	47289	47359	48344	47359
MATCHHILBERT	401815	357914	363288	358353	344742	370740	417821
MATCHSMALLESTX	138362	106530	73954	70796	57164	68377	107256
MATCHOMT	72590	57643	56258	51175	45958	50478	56514
MATCHSTR	45245	42892	42077	42613	43233	43218	43849

Research question Q_2. From Figure 4a, we can see that all the content-based measures implemented in COBALT (i.e., COBALT_{area}, $\text{COBALT}_{diagonal}$ and COBALT_{mixed}) with R-tree indexing are 4 to 8 times faster than their counterparts (i.e., $\text{COBALT}_{area(R)}$, $\text{COBALT}_{diagonal(R)}$ and $\text{COBALT}_{mixed(R)}$) deployed based on the RADON's space tiling indexing. For instance, the total runtime of COBALT_{mixed} is 195 milliseconds while the total runtime in $\text{COBALT}_{mixed(R)}$ is 849 milliseconds, which means that COBALT_{mixed} is 4.3 times faster than $\text{COBALT}_{mixed(R)}$. The slowest content-based measure of COBALT (i.e., the COBALT_{area}) is *on average* 4840 times faster than RADON. COBALT_{mixed} is up to 1.47×10^4 times faster than RADON, which is the best speedup COBALT has in comparison to all other algorithms. This shows clearly how efficient are the the content-based measures when it comes to the runtime, which clearly answers our second research question Q_2.

Research question Q_3. Based on the results of Table 5, we analysed the impact of using the content-based measures on the F-Measure. For discovering the `equals` relation based on MBBs of the original polygons, all algorithms achieved an F-Measure of 0.996. For the `intersects`, `contains` and `within` relations, the F-Measures were 0.852, 0.853 and 0.853, respectively. The `overlaps` relation was the most affected relation by using the MBBs. In the case of the NUTS×NUTS experiment for instance, using MBBs



(a) Total runtime of linking NUTS×NUTS.

(b) Total runtime of linking CLC×NUTS.

Figure 4. Runtime results of KG linking experiments.

instead of the original polygons as the input for discovering the `overlaps` relation resulted in 586 true positives (out of 790 or 74.47%), 26884 false positives, 4012426 true negatives (out of 4039310 or 99.33%) and 204 false negatives. In total, using MBBs classified 45 times more polygon pairs falsely as being overlapped than the true number of overlapping pairs. The high number of pairs that was correctly identified as not overlapping is caused by the indexing algorithm, which filters out a high percentage of non-intersecting pairs. The only relation where the content-based measures produce better F-Measures than both RADON$_{MBB}$ and GIA.NT$_{MBB}$ was the `touches` relation. Both RADON$_{MBB}$ and GIA.NT$_{MBB}$ were not able to detect the `touches` relation correctly in most cases as the intersection matrix of the MBBs depends heavily on the polygon shape. For instance, using both RADON$_{MBB}$ and GIA.NT$_{MBB}$ for discovering the `touches` relation for the NUTS×NUTS Experiment (again see Table 5) resulted in an F-Measure of 0.001, while COBALT$_{area}$ and COBALT$_{diagonal}$ achieved an F-Measure of 0.678 and 0.779, respectively. Both the area and diagonal measures benefited from the fact that there are 20150 pairs that `touch` each other but only 790 pairs that `overlap`. To summarise, by using the content-based measures we lose *on average* 32% of the F-Measure compared to the F-Measure of 1.0 produced by RADON or GIA.NT. This answers our third research question Q_3.

Research question Q_4**.** State-of-the-art approaches tend to use polygons simplification in order to speed up the link discovery of topological relations [22]. As part of our second set of experiments, we studied the trade-off between accuracy and efficiency by using content-based measures on the polygons' MBBs vs. using a simplified version of the original polygons. Based on the results of Table 5, the F-Measures of RADON$_{simp(0.05)}$, RADON$_{simp(0.1)}$ and RADON$_{simp(0.2)}$ for the relations `contains` and `within` were worse than all the results produced using MBBs of polygons. For instance, RADON$_{simp(0.05)}$, RADON$_{simp(0.1)}$, and RADON$_{simp(0.2)}$ achieved the F-Measures 0.7, 0.72 and 0.733, respectively. While using COBALT on the MBBs of the original polygons achieved an F-Measure of 0.853 for the `contains` and `within` relations. When using the polygon simplification algorithms, the F-Measure for the `equals` relation is 1.0 for RADON with simplified polygons when linking NUTS×NUTS. The content measures are able to achieve an F-Measure of 0.996 for this relation. From the aforementioned results, we can conclude that using content-based measures on the polygons' MBBs result in a better trade-

Table 5. F-Measure for linking NUTS×NUTS (all values rounded to three decimal places). The results of COBALT combined with RADON indexing are omitted, because the indexing does not change the accuracy.

Algorithm	`equals`	`intersects`	`contains`	`within`	`touches`	`overlaps`
RADON	1.000	1.000	1.000	1.000	1.000	1.000
$\text{RADON}_{\text{MBB}}$	0.996	0.852	0.853	0.853	0.001	0.041
GIA.NT	1.000	1.000	1.000	1.000	1.000	1.000
$\text{GIA.NT}_{\text{MBB}}$	0.996	0.852	0.853	0.853	0.001	0.041
$\text{RADON}_{\text{simp(0.2)}}$	1.000	0.916	0.733	0.733	0.177	0.068
$\text{RADON}_{\text{simp(0.1)}}$	1.000	0.953	0.721	0.721	0.199	0.064
$\text{RADON}_{\text{simp(0.05)}}$	1.000	0.980	0.700	0.700	0.209	0.061
COBALT_{area}	0.996	0.852	0.853	0.853	0.678	0.041
$\text{COBALT}_{diagonal}$	0.996	0.852	0.853	0.853	0.779	0.041
COBALT_{mixed}	0.996	0.852	0.853	0.853	0.001	0.041

Table 6. F-Measure for linking CLC×NUTS. All values rounded to three decimal places and - indicate the total absence of the relation in the result set. The results of COBALT combined with RADON indexing are omitted, because the indexing does not change the accuracy.

Algorithm	`equals`	`intersects`	`contains`	`within`	`touches`	`overlaps`
RADON	-	1.000	1.000	-	-	1.000
$\text{RADON}_{\text{MBB}}$	-	0.709	0.689	-	-	0.066
GIA.NT	-	1.000	1.000	-	-	1.000
$\text{GIA.NT}_{\text{MBB}}$	-	0.709	0.689	-	-	0.066
$\text{RADON}_{\text{simp(0.2)}}$	-	0.938	0.931	-	-	0.332
$\text{RADON}_{\text{simp(0.1)}}$	-	0.963	0.958	-	-	0.419
$\text{RADON}_{\text{simp(0.05)}}$	-	0.980	0.975	-	-	0.540
COBALT_{area}	-	0.709	0.689	-	-	0.066
$\text{COBALT}_{diagonal}$	-	0.709	0.689	-	-	0.066
COBALT_{mixed}	-	0.709	0.689	-	-	0.066

off between efficiency and accuracy than using a simplified version of polygons. We can see the same behavior also for our second linking task, i.e., CLC×NUTS, see results in Table 6. This clearly answers our fourth research question Q_4.

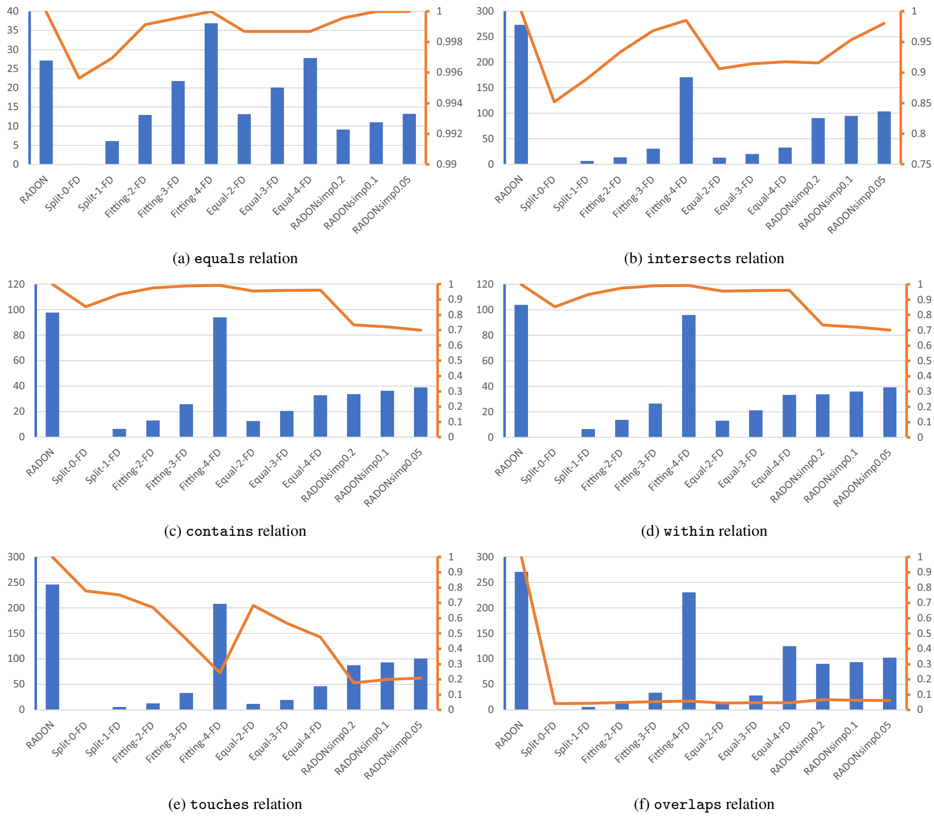
Research question Q_5 To answer Q_5, we conducted our *third set of experiments* by linking CLC against itself (i.e., CLC×CLC). For this experiment we implemented a parallelised version of COBALT, where we used $\{1,3,4,8\}$ thread(s). As shown in Table 7, all MBB based algorithms did not benefit from using multiple threads. Because the MBB-based algorithms are so fast, to the extent that the time needed for threads coordination is the same as the time saved by allocating the work to other threads. All of the MBB-based algorithms were able to finish linking CLC to itself in less than one hour. This is clearly answer our research question Q_5. On the other hand, multiple threads decreased the runtime of RADON and GIA.NT, because they use the intersection matrix that requires expensive computing which can take advantage of employing more threads. In particular, RADON is 6.31 times faster with 8 threads than with only one thread. All

Table 7. Runtime for linking CLC×CLC using a different number of threads. All runtimes are recorded in hours, where all values are rounded to three decimal places.

Algorithm	1 Thread	2 Threads	4 Threads	8 Threads
RADON	1179.489	686.007	340.827	186.913
RADON$_{\mathrm{MBB}}$	0.703	0.729	0.597	0.586
GIA.NT	1179.463	675.589	334.928	179.415
GIA.NT$_{\mathrm{MBB}}$	0.635	0.458	0.346	0.329
COBALT$_{area(R)}$	0.710	0.670	0.583	0.573
COBALT$_{diagonal(R)}$	0.539	0.564	0.513	0.505
COBALT$_{mixed(R)}$	0.494	0.550	0.509	0.503
COBALT$_{area}$	0.209	0.215	0.186	0.192
COBALT$_{diagonal}$	0.190	0.195	0.175	0.183
COBALT$_{mixed}$	**0.179**	**0.190**	**0.171**	**0.180**

of the content measures with R-tree indexing are at least three times faster than RADON with MBBs.

Research question Q_6**.** To study the trade-off between accuracy and efficiency when we apply our polygons splitting strategies (i.e., the equal split and the fitting split strategies) before applying the content measures, we conducted our *last set of experiments*. In particular, we are interested in comparing COBALT with the two splitting strategies to other approximation algorithms (i.e., polygon simplification). For this experiment we compute the topological relations for NUTS×NUTS. We benchmarked both split strategies defined in Section 3 against RADON and the combination of RADON and polygon simplification as we did in the previous experiments. The splitting algorithms are combined with the diagonal-based content measure. We dubbed our first splitting strategy (depicted in Figure 3a) as EQUAL-t-FD and the second splitting strategy as FITTING-t-FD (depicted in Figure 3b) with t being the number of recursive splits (We used 0 to 4 recursive splits) and FD being the diagonal-based content measure. As both split strategies produce the same result for 0 and 1 recursive splits, we only each of them once as SPLIT-0-FD and SPLIT-1-FD. For the `equals` relation (see Figure 5a), the diagonal content measure function achieved an F-Measure of 0.996 before applying the splitting algorithm on the polygons. The fitting split strategy with 3 and 4 recursive splits (i.e., FITTING-3-FD and FITTING-4-FD) achieved the best accurate results. On the other hand, the polygon simplification algorithms were all able to achieve perfect results in less time than both FITTING-3-FD and FITTING-4-FD. The diagonal content measure however achieved a high F-Measure of 0.996 while being over 100 times faster than the simplification algorithms. For the `intersects` relation (see Figure 5b), SPLIT-0-FD achieved an F-Measure of 0.852 without splitting polygons. The *fitting-split strategy* has a better accuracy for the `intersects` relation than the *equal-split strategy* for each t but with increased runtime. When compared FITTING-3-FD to EQUAL-4-FD we also notice that FITTING-3-FD is both faster and more accurate than EQUAL-4-FD. FITTING-3-FD is three times faster than the simplification algorithms and is only slightly worse in accuracy than the RADON$_{\mathrm{simp}(0.05)}$ algorithm, but better than RADON$_{\mathrm{simp}(0.1)}$ and RADON$_{\mathrm{simp}(0.2)}$. For the `contains` and `within` relation (see Figure 5c and 5d), the content measures without splitting were able to achieve an F-Measure of 0.853, which was already better



(a) `equals` relation (b) `intersects` relation

(c) `contains` relation (d) `within` relation

(e) `touches` relation (f) `overlaps` relation

Figure 5. Runtime in seconds (blue) and F-Measure (orange) results for linking NUTS $\times$ NUTS.

than the simplification algorithms. By splitting the polygons, we were able to achieve a higher F-Measure.In particular, the FITTING-3-FD was 99% accurate while it used only 26.4% of the runtime RADON needs to compute the `contains` relation. This indicated that FITTING-3-FD was the strategy with the best runtime/accuracy trade-off. The `overlaps` relation (see Figure 5f) is a relation that cannot be accurately detected by the content measures or the polygon simplification algorithms. In this case splitting the polygons has a positive effect on the accuracy, but it is still too low to be usable with all F-Measures being smaller than 0.07. For the `touches` relation (see Figure 5e), the diagonal content measure was able to achieve a higher F-Measure (0.779) without splitting. This happens because the diagonal content measure focuses on recall rather than precision and by splitting the polygons there are more cases where parts from the two polygons are overlapping. Therefore, the splitting reduces accuracy and the normal diagonal content measure function should be used for the `touches` relation. Overall, splitting polygons is good way to improve the accuracy of COBALT for the spatial relations `intersects`, `contains`, and `within`. Our experiments show that the FITTING-T-FD achieves a higher F-Measure than the EQUAL-T-FD algorithms for each respective T but also have a higher runtime. By using our splitting technique, we could guarantee to finish a linking task in a predetermined amount of time while also fully utilizing the time to maximize the accuracy of the result. This answers our research question Q_6.

5. Related Work

In last years many algorithms have been proposed to address both the efficiency and accuracy of link discovery in general and link discovery over geospatial RDF KGs in particular. For instance, *Ngonga Ngomo* [1] computes the distance between geographical items using the Hausdorff distance. *Sherif et al.* [23] provide a review of 10 point-set distance metrics for link discovery. SILK [24] computes topological relations in accordance with the DE-9IM standard based on the MultiBlock. To compute topological relations between geographical resources quickly and accurately, RADON [5] offers an indexing technique coupled with space tiling approach. While RADON computes the intersection matrix for each relation between a pair of geometries repeatedly, RADON2 [6] caches the computed intersection matrix and reuses it whenever it is possible. In GIA.NT [7], *Papadakis et al.* have adapted RADON's indexing. In particular, instead of calculating the estimated total *hypervolume* to decide which dataset to index, the authors simply index the first dataset using a grid approach. In DORIC [8], *Jin et al.* the relation computation problem is optimized by using existing links to infer new links. For instance, in case A equals B and A equals C, DORIC infers that B equal C. *Ahmed et al.* [11] studied the effect of simplifying the resources' geometries on the runtime and F-Measure of link discovery approaches over geospatial KGs. However, our approach computes topological relations using content measures as defined in [12] instead of computing the DE-9IM intersection matrix. Accuracy has received a considerable attention from the research society of link discovery. For instance, algorithms such as RADON [5], RADON2 [6], GIA.NT [7], and DORIC [8] achieve an F-Measure of 1, while algorithm such as the ones in [11] and our presented algorithm here (COBALT) scarify the accuracy in favour of efficiency.

6. Conclusion & Future Work

In this paper, we propose COBALT, an approach for topological relation discovery. COBALT combines the R-tree indexing with the content-based measures in order to scale up the topological relations discovery process. Based on our experiments, COBALT is able to achieve a speed up of up to 1.47×10^4 over state-of-the-art algorithms. On the other side, we also study the impact of applying our proposed approach on the accuracy of the generated links. In order to optimize COBALT, we propose two polygon splitting strategies. Without applying our splitting strategies, COBALT achieves an F-Measure between 70% and 90%. By applying our proposed splitting strategies, the F-Measure of COBALT is improved to up to 99%. In future work, we aim to improve the accuracy of COBALT by incorporating more non-spacial information into the linking process. In particular, we will consider information regarding the type, location, description, and name of the resources represented by the polygons in the linking process.

Acknowledgements

This work has been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL (grant no NW21-059D) and the European Union's Horizon Europe research and innovation program (grant no 101070305). The authors gratefully acknowledge the funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC2).

References

[1] Ngonga Ngomo AC. ORCHID - Reduction-Ratio-Optimal Computation of Geo-Spatial Distances for Link Discovery. In: Proceedings of ISWC 2013; 2013. p. 395-410.

[2] Auer S, Lehmann J, Hellmann S. LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In: Bernstein A, Karger DR, Heath T, Feigenbaum L, Maynard D, Motta E, et al., editors. The Semantic Web - ISWC 2009. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. p. 731-46.

[3] Berners-Lee T. Linked-data design issues; 2009. W3C design issue document. Available from: https://www.w3.org/DesignIssues/LinkedData.html.

[4] Zhang C, Li W. The roles of web feature and web map services in real-time geospatial data sharing for time-critical applications. Cartography and Geographic Information Science. 2005;32(4):269-83.

[5] Sherif MA, Dreßler K, Smeros P, Ngomo ACN. Radon–rapid discovery of topological relations. In: Thirty-First AAAI Conference on Artificial Intelligence; 2017. .

[6] Ahmed AF, Sherif MA, Ngomo ACN. Radon2: a buffered-intersection matrix computing approach to accelerate link discovery over geo-spatial rdf knowledge bases. In: Ontology Matching: OM-2018: Proceedings of the ISWC Workshop; 2018. p. 197.

[7] Papadakis G, Mandilaras G, Mamoulis N, Koubarakis M. Progressive, Holistic Geospatial Interlinking. In: Proceedings of the Web Conference 2021. WWW '21. New York, NY, USA: Association for Computing Machinery; 2021. p. 833–844. Available from: https://doi.org/10.1145/3442381.3449850.

[8] Jin X, Eom S, Shin S, Lee KH, Hong C. DORIC: discovering topological relations based on spatial link composition. Knowledge and Information Systems. 2021 Oct;63(10):2645-69. Available from: https://doi.org/10.1007/s10115-021-01603-2.

[9] Clementini E, Di Felice P, van Oosterom P. A small set of formal topological relationships suitable for end-user interaction. In: Abel D, Chin Ooi B, editors. Advances in Spatial Databases. Berlin, Heidelberg: Springer Berlin Heidelberg; 1993. p. 277-95.

[10] Egenhofer MJ, Mark DM, Herring J. The 9-Intersection: Formalism and Its Use for Natural-Language Spatial Predicates (94-1). UC Santa Barbara: National Center for Geographic Information and Analysis; 1994.

[11] Ahmed AF, Sherif MA, Ngomo ACN. On the Effect of Geometries Simplification on Geo-spatial Link Discovery. Procedia Computer Science. 2018;137:139-50. Proceedings of the 14th International Conference on Semantic Systems 10th – 13th of September 2018 Vienna, Austria. Available from: https://www.sciencedirect.com/science/article/pii/S1877050918316193.

[12] Godoy F, Rodríguez A. Defining and Comparing Content Measures of Topological Relations. Geoinformatica. 2004 dec;8(4):347–371. Available from: https://doi.org/10.1023/B:GEIN.0000040831.81391.1d.

[13] Balasubramanian L, Sugumaran M. A State-of-Art in R-Tree Variants for Spatial Indexing. International Journal of Computer Applications. 2012 03;42:35-41.

[14] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching. SIGMOD Rec. 1984 jun;14(2):47–57. Available from: https://doi.org/10.1145/971697.602266.

[15] Kamel I, Faloutsos C. On Packing R-Trees. In: Proceedings of the Second International Conference on Information and Knowledge Management. CIKM '93. New York, NY, USA: Association for Computing Machinery; 1993. p. 490–499. Available from: https://doi.org/10.1145/170088.170403.

[16] Leutenegger S, Lopez M, Edgington J. STR: A Simple and Efficient Algorithm for R-Tree Packing; 1997. p. 497-506.

[17] Lee T, Lee S. OMT: Overlap Minimizing Top-down Bulk Loading Algorithm for R-tree. In: CAISE Short paper proceedings. vol. 74; 2003. p. 69-72.

[18] Kamel I, Faloutsos C. Hilbert R-Tree: An Improved R-Tree Using Fractals. In: Proceedings of the 20th International Conference on Very Large Data Bases. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1994. p. 500–509.

[19] Beckmann N, Kriegel HP, Schneider R, Seeger B. The R*-tree: an efficient and robust access method for points and rectangles. In: SIGMOD '90; 1990. .

[20] Büttner G, Kosztra B. CLC2018 Technical Guidelines. European Environment Agency; 2017. Available from: https://land.copernicus.eu/user-corner/technical-library/clc2018technicalguidelines_final.pdf.

[21] Douglas DH, Peucker TK. ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE. Cartographica: The International Journal for Geographic Information and Geovisualization. 1973;10:112-22.

[22] Ahmed AF, Sherif MA, Ngomo ACN. On the Effect of Geometries Simplification on Geo-spatial Link Discovery. In: SEMANTiCS 2018 - Research Track. SEMANTiCS '18; 2018. Available from: http://svn.aksw.org/papers/2018/SEMANTICS_GeoSimp/paper/public.pdf.

[23] Sherif MA, Ngomo ACN. A systematic survey of point set distance measures for link discovery. Semantic Web Journal(Cited on page 18). 2015.

[24] Smeros P, Koubarakis M. Discovering Spatial and Temporal Links among RDF Data. In: LDOW@ WWW; 2016. .

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230014

Classification of Linking Problem Types for Linking Semantic Data

Raphaël CONDE SALAZAR [a,1], Clément JONQUET [a,b] and Danai SYMEONIDOU [a]
[a] *MISTEA, University of Montpellier, INRAE & Institut Agro, France*
[b] *LIRMM, University of Montpellier & CNRS, France*
ORCiD ID: Raphaël Conde Salazar https://orcid.org/0000-0002-6926-5299, Clément Jonquet https://orcid.org/0000-0002-2404-1582, Danai Symeonidou https://orcid.org/0000-0003-1152-5200

Abstract. As the number of RDF datasets published on the semantic web continues to grow, it becomes increasingly important to efficiently link similar entities between these datasets. However, the performance of existing data linking tools, often developed for general purposes, seems to have reached a plateau, suggesting the need for more modular and efficient solutions. In this paper, we propose –and formalize in OWL– a classification of the different Linking Problem Types (LPTs) to help the linked data community identify upstream the problems and develop more efficient solutions. Our classification is based on the description of heterogeneity reported in the literature –especially five articles– and identifies five main types of linking problems: predicate value problems, predicate problems, class problems, subgraph problems, and graph problems. By classifying LPTs, we provide a framework for understanding and addressing the challenges associated with semantic data linking. It can be used to develop new solutions based on existing modularized tools addressing specific LPTs, thus improving the overall efficiency of data linking.

Keywords. Data linking, Semantic web, Linking Problem Types, Classification

1. Introduction

For more than twenty years, important work has been going on for the development of the semantic web [1] with the aim of sharing data online and facilitating access by machines to human knowledge. In this approach, Linked Open Data (LOD) promotes the sharing and reuse of royalty-free datasets, based on the semantic web model and tools, such as the RDF and OWL representation languages. But while the number of datasets available as LOD is increasing every year, a new challenge must be met: data linking. Indeed, in order to maximize the knowledge from a resource, agents browsing these datasets must be able to link two resources designating the same thing but identified by distinct identifiers (URIs) within each of datasets. For example, a prominent actor might venture into politics, resulting in their inclusion and description within separate knowledge bases for cinema and politics. In order to write his biography, the two URIs, generally distinct, which identify this same person must be linked-back by an equivalence link such as the semantic relation `owl:sameAs` whose uses are described in [2,3,4,5].

[1]Corresponding Author: Raphaël Conde Salazar, raphael.condesalazar@online.fr

The very fact that data linking is based on *similarity* is problematic if one considers that it is a subjective notion that is difficult to grasp in a formal way. Indeed, similarity does not express exact likeness but a close resemblance or similitude for which automated processing must then be parameterized in an equally subjective manner. For example, let us take the descriptions in two datasets of two distinct persons homonyms by their first and last names. Should we consider semantically that they are the same person if their date of birth is identical or should we consider the low probability that these two homonyms were born on the same day. In addition to the difficulty of setting up decision trees, the *heterogeneity* of the datasets is also a major obstacle to data linking. Indeed, taking the previous example, one would like to choose one or more characteristics that would uniquely identify the two persons such as their social security number, but would be embarrassed if this property were described using two similar, yet formally different, predicates (e.g., `hasSocialSecurityNumber` and `hasSSN`) and/or if the value of this property was presented in a different format (e.g., the number 1880475114782 and the literal "1-88-04-75-114-782"). The search for similarity between two (ontological) entities is therefore strongly impacted by the different semantic, lexical, or structural heterogeneities that can be obtained from the design of the datasets given the constraints imposed by RDF. The non-respect of good practices such as the non-use of language tags for labels, or serialization errors such as the presence of duplicate identification keys, can also reinforce these heterogeneity problems. All these heterogeneities make data linking based on similarity more complex and tedious, and also require the intervention of experts.

Several data linking tools have been developed according to different strategies [6], and are confronted during benchmarking campaigns such as the Ontology Alignment Evaluation Initiative (OAEI) [7], an annual event to evaluate ontology alignment and data linking tools. In general, the competing tools offer very generic solutions composed of several modules in an attempt to resolve a maximum number of types of heterogeneity presented in these tests. Although high, the maximum efficiency of these tools seems to have stabilized in recent years without reaching a fully reliable ideal solution as pointed out by Algergawy et al. [8] and Pour et al. [9,10] in the conclusions of their presentations of the OAEI benchmark results for the years 2019, 2020 and 2021.

In this paper, we suggest and anticipate a novel technique for data linking, which involves creating profiles for pairs of datasets and utilizing machine learning algorithms to recommend which modular solutions would be best suited to these profiles for the linking task. This approach deviates from the conventional incremental methods currently used and takes advantage of already existing data linking tools and datasets. To establish these profiles, all the problems that can be encountered when linking two data sets must be identified. We therefore propose a classification of the different types of Linking Problem Types (LPTs) that can be encountered during semantic data linking. This classification of LPTs, also formalized in OWL, will be publicly accessible to the community for inclusion in automatic tools and future improvements. To the best of our knowledge, there is no such a formalized classification of the types of semantic data binding or linking problems and we believe that this is an impediment to a fully automated treatment of data linking, especially with new machine-learning based approach coming.

The rest of the article is organized as follows: we present related work in Section 2. Then, we present our vision of similarity when linking two RDF entities in Section 3, as well as the methodology used to build our classification from the different types of heterogeneities coming from the data linking corpus in Section 4. Section 5, details our resulting classification and its formalization in OWL. Finally, we discuss perspectives and conclude our work in Section 6.

2. Related work

Data linking has been defined for example as:“ the task of establishing typed links between entities across different RDF datasets via the help of automatic link discovery systems .” [11]. In [12], the authors distinguish two approaches to data linking: “(i) A similarity-based approach in which the more similar two resources are, the more likely they are to be linked; (ii) A key-based approach in which a key determines the identity of a resource: two resources with the same key must be linked”. In this paper, we define data linking as the task of establishing similarity or hierarchical relationships between distinctly identified entities in two different semantic datasets.
Here is a summary of some common data linking techniques:

- *Deterministic linking* involves linking dataset records based on unique identifiers (such as the social security number in the previous example) or other unique identifiers that allow for a one-to-one match between entities in different datasets. This method is considered the most accurate and efficient but requires the use of common unique identifiers across datasets which may not exists or may be hard to identify.
- *Probabilistic linking* involves matching entities based on non-unique identifier properties such as names, addresses, and dates of birth. The technique calculates the probability that two records refer to the same entity.However, determining a unique key on the basis of several pairs of properties and values remains an arduous task.Work has been carried out for the automated determination of these keys [13,14,15]. Probabilistic linking is useful when unique identifiers are not available, but it is less accurate than deterministic linking.
- *Rule-based linking* involves defining rules that specify the conditions for linking records between datasets [16,17]. For example: two records match if they have the same name and address. Rule-based linking can be useful when there is a high degree of certainty about the conditions for linking records.
- *Knowledge graph embedding* involves representing relationships as translations in the embedding space [18]. Graphs are transposed into vector spaces because the latter offer a wider range of tools for mathematical and statistical processing. This means that technologies such as machine learning can be applied more easily to these graphs. This is one of the most recent techniques in the field of data liking, but despite the many advantages it has over other, more traditional techniques, it still has limitations, such as the fact that this method does not hold up well when the relational paths are long or complex [19].

3. What it is for two RDF resources to be similar?

According to the Larousse French dictionary, similar things are defined as follows: “A set of things that can, in a certain way, be assimilated to each other”. In the following, we review how this definition may be applied to RDF resources belonging to different datasets by trying to clarify this notion “in a certain way”.

Harispe et al. [20] say:“Similarity assessment must therefore not be understood as an attempt to compare object realisations through the evaluation of their properties, but rather as a process aiming to compare objects as they are understood by the agent which estimates the similarity (e.g., a person, an algorithm). The notion of similarity therefore only makes sense according to the consideration of a partial (mental) representation on which the estimation of object similarity is based". We agree with this last quote and the fact that the simple observation of common characteristics between two entities is not enough to make them similar. Indeed, if we are interested in two person whose descriptions indicate that their height is precisely 1m78 this characteristic would not be sufficient to allow us to affirm that they are the same person. Nevertheless, the observation of an entity's characteristics is primordial in a similarity search, because one or a set of characteristics can make an entity unique through its description (e.g., social security number, or the set of first name, last name, date and place of birth, eye color and postal address). The observation of certain characteristics can also invalidate our search for similarity (e.g., we will not try to compare two person if their description reports that one has blue eyes and the other brown). This use of an entity's characteristics in a similarity search implies de facto that we can compare the same characteristic and its value in a similar way. For our objective of setting up a classification of LPTs, we are thus simplistically focused on the comparison of one and the same characteristic of two RDF resources supposed to represent the same entity through its description in two distinct RDF datasets which can also not be obvious as we will see.

Let two RDF triples (see Fig. 1) then, S and S' are considered similar if they share a common characteristic, which implies that P and P' are similar as well as O and O'.

Figure 1. Two RDF triples.

We intuitively identified four types of issues for similarity between S and S':

1. If P=P', then the issue is in establishing similarity between O and O' (see Fig. 2.a). The problem can be linguistic or structural when the object is a literal. For examples, the value of the property `hasForQualification` of a person can be described via the literal "coach operator" in one dataset and "bus driver" in another. Or the value of the property `dateOfBirth` of a person can be described via the literal "24 march 2023" in one dataset and "2023-03-24" in another due to a difference in the date format.

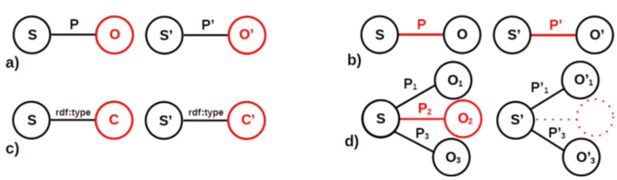


Figure 2. **a)** First intuitive issue: Objects are different while subjects and predicates are the same.
b) Second intuitive issue: Predicates are different while subjects and objects are the same.
c) Third intuitive issue: The subjects are identical but belong to different classes.
d) Fourth intuitive issue: Missing characteristics from the description of one of the subjects.

2. If O=O', then the issue is in establishing similarity between P and P' (see Fig. 2.b). The problem can be linguistic, structural or semantic. At the linguistic level, for example, the professional qualification of a person could be expressed through the

predicate `hasForQualification` in a dataset mostly in English whereas the equivalent predicate may be `aPourQualification` in another dataset mostly in French. At the structural level, for example, a characteristic of a person like his date of birth can be represented by the property `dateOfBirth` in a dataset and this same characteristic by the three properties `monthOfBirth`, `dayOfBirth` and `yearOfBirth` in another dataset. At the semantic level, for example, the name of a person can be described by the relation `foaf:name` from the FOAF vocabulary[2] in a dataset and by the relation `vcard:fn` of the vCard ontology[3] in another one.
3. If predicates indicates that the subject is the instance of a class (e.g., `rdf:type`), then the issue is in establishing similarity between the types of an entity C and C' (see Fig. 2.c). The problem may be one of terminology or specialisation/generalisation. For example, a person can belong to two different subclasses of a given class, it can be an instance of the class `Person` and an instance of the class `Actor`, both subclasses of the class `Human`.
4. Issue in establishing similarity of entities when a property of S is absent for S' (see Fig. 2.d). For example, a person can be described with his name, date of birth and social security number in one dataset and only with his name and date of birth in another.

We find these four issues originate in the diversity of values, structure, and logic used in the development of the compared datasets. These types of problems are known and arise from the flexibility of RDF which, as we have seen in the previous examples, does not impose any constraints on the data, just formalize how to encode them. These types of terminological and structural problems have been reported in the context of XML exploitation and are described in the literature under the term heterogeneity. The RDF syntax (i.e., RDF/XML) is based on XML and therefore inherits these heterogeneity problems. We will then use these four issues to initialize our classification of LPTs.

4. Methodology used for the construction of the LPTs classification

We will now confront the four issues previously discussed with the different types of heterogeneities reported in the data linking literature. To achieve this objective, we start by building a small corpus of articles about data linking, then from this corpus, we keep only the articles dealing with the heterogeneities that can be encountered during semantic data linking.

4.1. Analysis of data linking literature

We did a systematic review of data linking literature in order to compile an exhaustive list of LPTs. Figure 3 illustrates our approach: we started from a very specific term (e.g., "OAEI") with which we performed a first bibliographic search with Google Scholar and Web of Science to create a corpus of research papers (as PDF documents). Then the extracted corpus is fed to a text mining tool called Gargantext[4], to obtain a list of words that are considered statistically relevant to the topic covered by this corpus. See, for example, Table 1 for the results obtained for the expression "OAEI". This list of all terms is then re-injected into a bibliographic search whose articles obtained are again re-injected

[2]Friend Of A Friend vocabulary. http://www.foaf-project.org/
[3]vCard Ontology for describing People and organizations. https://www.w3.org/TR/vcard-rdf/
[4]A web platform for text-mining. https://gargantext.org of the Institute of Complex Systems (Paris).

Table 1. The first fifteen compound multi-words and their occurrences extracted by data mining with the keyword "OAEI" in a corpus of two hundred documents.

Multi-word extracted	Occurrence
ontology matching	59
semantic web	42
ontology mapping	39
ontology alignment	38
different ontology	36
data sources	16
cheminform abstract	16
ontology alignment evaluation initiative	16
open data	15
schema matching	14
schema matcher	14
semantic interoperability	12
matching process	11
large ontologies	11
similarity measures	10

into the text mining process to enter a virtuous circle. We stoped the process when we considered the list of terms extracted by the text mining stops evolving.

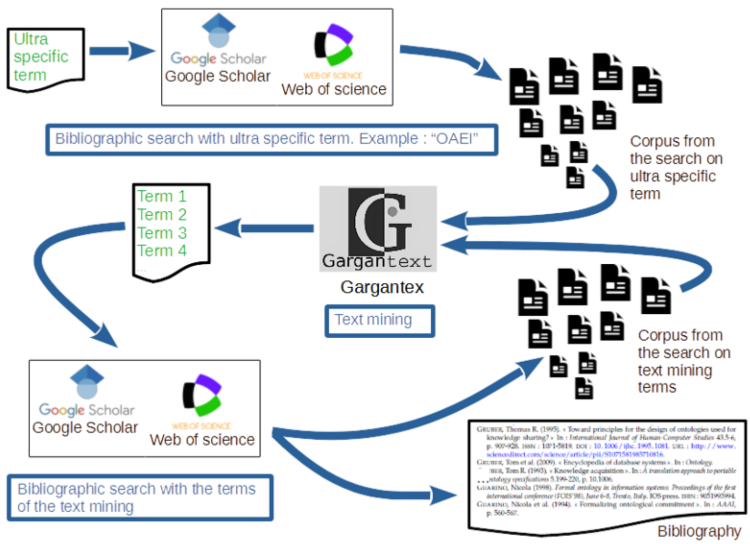


Figure 3. Methodology for bibliographic search and enrichment.

We have thus obtained a first set of relevant documents in the field of data linking and its different techniques. It is from this first corpus that we will subsequently extract a list of five articles dealing with the problems of heterogeneity.

4.2. Review of articles addressing different heterogeneity issues

We classify the different forms of heterogeneity found in the literature according to our four issues in order to continue our classification of LPTs. To do so, we manually reviewed the articles dealing with heterogeneities compiled in the corpus explained above. These articles are either about Instance Matching (IM) where one consider assertions (notion of instance) or about Ontology Matching (OM) where one deal with the reconciliation of models (notion of classes). Although distinct, these two domains are complementary in the execution of data linking tasks. We have finally selected the following five articles for their relevance and their global vision on the subject of heterogeneity:

- Klein [21] proposes a classification of different heterogeneities encountered in the combined use of independently constructed ontologies. This work identifies three main families of heterogeneities:
  - ∗ Heterogeneity related to practice (e.g., non application of language tags, input errors, duplicates).
  - ∗ Heterogeneity linked to the mismatch of languages used to express these ontologies. At this level, a distinction is made between heterogeneities related to the linguistic level (e.g.,syntax, representation, semantics and expressivity) and those related to the ontological level (e.g., paradigm, concept description, coverage of model, synonymy).
  - ∗ Heterogeneity related to the versioning of one or more of the ontologies involved.
- Bergman [22] addresses the issue of resolving semantic heterogeneities in the context of using the semi-structured XML language (based largely on the work of Pluempitiwiriyawej and Hammer [23]) and, by extension, RDF and ontology representation languages like OWL. He considers that even within an identical domain there will always be different "world views" as long as independent teams create ontologies due to the flexibility of semi-structured schemas. Moreover, during serialization, XML files and ontologies can be confronted with syntax or structure problems. This work identifies four categories of causes for these heterogeneities:
  - ∗ Heterogeneity related to structure. This occurs when the schemas of the sources that represent related or overlapping data do not match (e.g., first and last name aggregation).
  - ∗ Heterogeneity related to domain. This occurs when the semantics of the data sources are different (e.g., Different scales and units of measurement).
  - ∗ Heterogeneity related to data. This occurs when there are discrepancies between the values of similar or related data (e.g., spelling mistakes).
  - ∗ Heterogeneity related to language.This occurs when there are differences in the encoding and use of different languages (e.g., Use of French and English).

  Bergman estimates there are more than forty discrete categories of heterogeneity. As our work is focused on RDF datasets, some of the heterogeneities described in the context of the use of XML seemed irrelevant (e.g., the notion of element order which is non-existent in RDF). Of the forty or so heterogeneities presented, we have selected twenty-six which fall into the four main categories.
- Euzenat and Shvaiko's work [24] is related to OM rather than IM; but still bring in an interesting analysis of heterogeneities that we can apply to data linking. They consider the following four main types of heterogeneities:
  - ∗ Syntactic heterogeneity: ontologies are expressed in different representation languages.
  - ∗ Terminological heterogeneity: ontologies have variations in naming objects (car vs. automobile).
  - ∗ Conceptual heterogeneity: ontologies have differences in modeling choices for the same domain. They can be differences in coverage, granularity or perspective.

  * Semiotic heterogeneity: ontologies describe the same thing (e.g., a sharp metal blade with a handle) that people/users will interpret differently depending on the context (a knife can be a weapon or a kitchen utensil).
- Achichi et al. [25] pragmatically classify the heterogeneities encountered by the designers of data linking tools as:
  * Value dimension: for heterogeneity problems at the level of terminology, language used and distinction between datatype properties and object properties.
  * Ontological dimension: for the problems of heterogeneity of vocabularies, structures, property depth, description and key.
  * Logical dimension: for class and property heterogeneity problems.
  * Data quality dimension: for the problems of transgression of good practice, heterogeneity of value type or non-updated dataset.
- Assi et al. [26] address the issue of IM. They introduce the scalability problem when it comes to IM on large datasets. Plus, they classify the heterogeneities as:
  * Value heterogeneity: gathering the notions of multilingualism, data format and data quality.
  * Structural heterogeneity: gathering the notions of vocabulary heterogeneity, predicate level and predicate granularity.
  * Logical heterogeneity: gathering the notions of hierarchical variation.

We found many similarities between these different heterogeneities, both in terms of organization by level and detail, but also many differences. We justify this diversity by the fact that the domains covered are not necessarily identical. For example, instance matching and ontology matching and because the levels of detail of each studies is different. Through these five articles, we were able to identify 69 descriptions of heterogeneity (See Table 2).

Table 2. Number of heterogeneity descriptions by authors.

Author(s)	Number of heterogeneity descriptions	Reference
Klein	11	[21]
Bergman	26	[22]
Euzenat and Shvaiko	6	[24]
Achichi et al.	13	[25]
Assi et al.	13	[26]

In order to better refer to them, we established a summary fact-sheet for each type of heterogeneity encountered, identified by a token. The tokens have been colored according to the authors who report them, as shown in Figure 4 presenting one of these summary fact-sheet.

4.3. *An iterative methodology*

In Section 3, we introduced four issues to evaluate how well they correspond to the various types of heterogeneities discussed in the paper corpus. To incorporate these heterogeneities into our new classification, we conducted manual clustering iterations. An example of this process is illustrated in Figure 5. In each iteration, we categorized the tokens into different themes based on their authors' descriptions to refine our classification. Some tokens were found in multiple clusters in subsequent iterations –as in the case where Assi et al. [26] mentioned that “the incorrectness simply refers to the

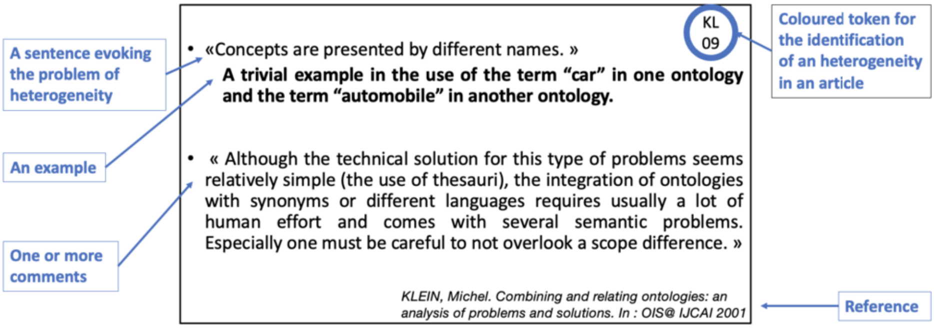


Figure 4. fact-sheet on synonymy problems of concept names according to Klein.

data typographical errors"– which can affect the value of a predicate as well as on the predicate itself. As a result of the first iteration, we were unable to classify some heterogeneities, which prompted us to establish a fifth primary level called "Problem at graph level". For example, when Achichi et al. [25] talks about key heterogeneity: "a property used to provide individual identifiers specific to a dataset, for example the identifiers of bibliographic entries in two libraries. In both cases, the values of these key properties are not comparable from one dataset to another." We could not classify this problem of heterogeneity within any of our four initial issues.

After four iterations, we arrived at a final classification that addresses all heterogeneity issues, organized into five primary levels based on the heterogeneity descriptions found in the literature.

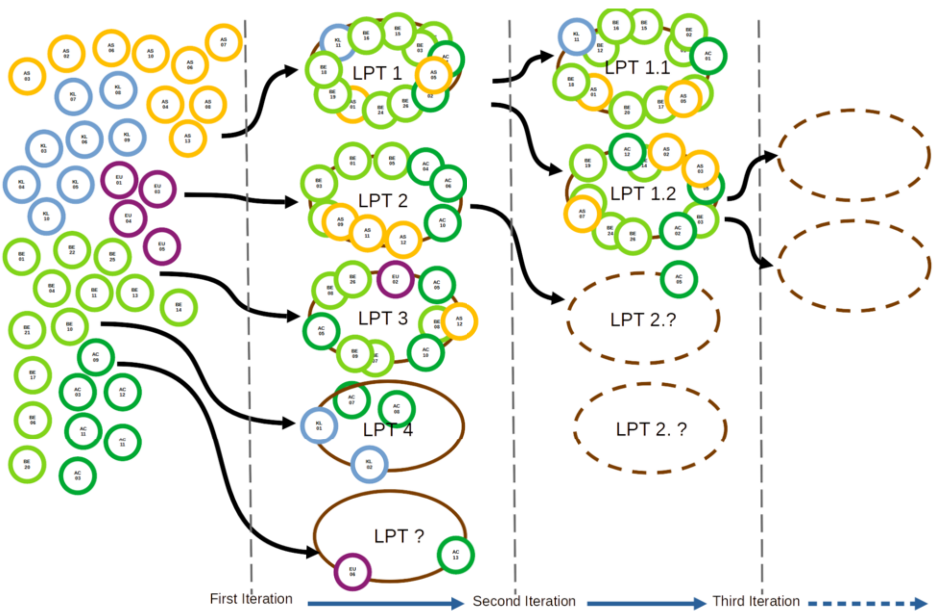


Figure 5. Schema of our iterative approach to develop the classification of LPTs, based on the reported heterogeneities in the selected articles. Each token in the figure represents a heterogeneity described in an article and is associated with a specific author and colour-coded accordingly.

5. Results

5.1. A classification for Linking Problem Types (LPTs)

We present here the results of the classification process previously explained. At the first hierarchy level, we find the four intuitive groups of problems to which we have added another group (i.e.,“Problem at graph level”) to capture problems related to the nature of the graphs (see Figure 6). In this Figure and the following ones, the colored pie charts represent the different distributions of heterogeneities described by each of the previously selected authors.

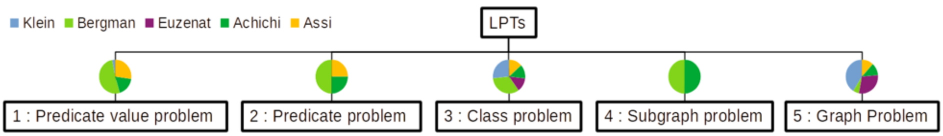


Figure 6. First level of the hierarchical classification of Linking Problem Types (LPTs).

5.1.1. Predicate value problems

Predicate value problems can be divided into terminological problems on one side and structural problems on the other (Figure 7). At the terminological level, the classification extends over three levels of granularity expressing at the finest level the problems of synonymy, homonymy or language reported in the literature mainly by Bergman [22] and Assi et al. [26].

A problem that would fall within the scope of LPT 1.1.2.5 would be, for example, a pair of datasets in which the data would be inconsistent (e.g., New York City would be described with 8,804,190 inhabitants on one side and 8,800,000 inhabitants on the other). For LPT 1.1.3, an example would be, a dataset pair in which literals do not have language labels (e.g., ,“barbecue” instead of “barbecue@en”), which would prevent automatic determination of the label language. For LPT 1.2.2, an example would be, a dataset pair in which the city of New York is represented as the object of a triple by the literal ”New York”@en on the one hand, and by its URI, `https://www.wikidata.org/wiki/Q60` pointing to the corresponding Wikidata page on the other.

5.1.2. Predicate problems

Predicate problems can be divided into predicate terminological problems, predicate structural problems (as predicate value problems) and predicate vocabulary problems (see Fig. 8).

An issue within the scope of LPT 2.1.5 would be, for example, a pair of datasets where the same predicate has a typing error (e.g., `hasPopulation` on one side and `hasPupoltion` on the other). For LPT 2.2.3, an example would be, a dataset pair in which a extra node (which can be a blank node) must be inserted or deleted in order to retrieve the same information (e.g., `New York hasNikeName Big Apple` on one side and `New York isCalled _b1 hasNikeName Big Apple ; New York isCalled _b1 hasAcronym NYC` on the other). For LPT 2.3 an example would be, a dataset pair which would express the same information using predicates from different vocabularies (e.g., `foaf:name` on one side and `rdfs:label` on the other).

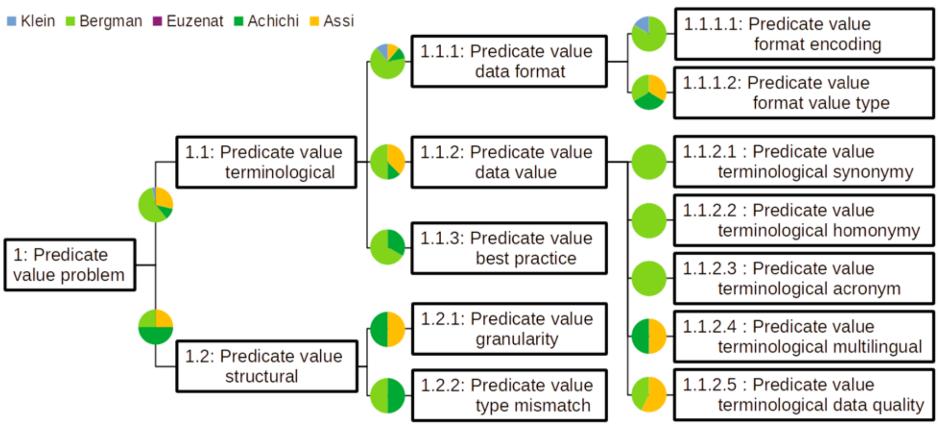


Figure 7. Part of the classification of LPTs: Predicate value problems.

At the level of granularity, we notice that the final levels of the classification are only described by a few authors. For example, only Bergman describes in detail the heterogeneities associated with terminological synonymy, homonymy and acronymy.

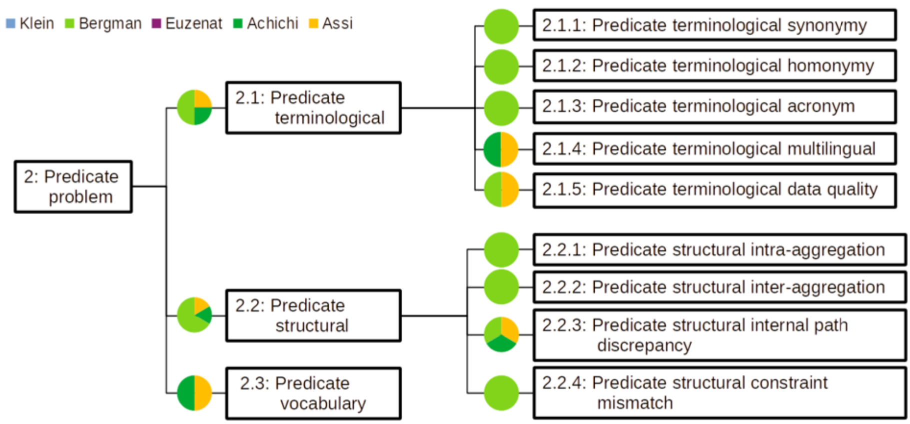


Figure 8. Part of the classification of LPTs: Predicate problems.

5.1.3. Class problems

Class problems can be divided into class terminological problems on one side and specialization/generalization on the other (see Fig. 9). We find a clustering around terminological problemes, which seems normal, if one consider it is a special case of a predicate value problem. Another grouping appears around the specialization/generalization problem more specific to the class domain reported by Klein [21], Bergman [22] and Achichi et al. [25].

For LPT 3.1.4 an example would be, a dataset pair in which there are variations in names for the same concept (e.g., `Paper` on one side and `Article` on the other). A example for LPT 3.2 would be, a dataset pair in which more general or specific concept are ised (e.g., `Phone` on one side and `HomePhone` or `Smartphone` on the other).

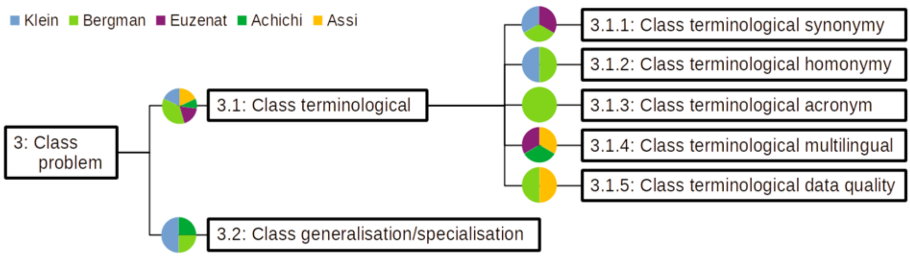


Figure 9. Part of the classification of LPTs: Class problems.

5.1.4. *Subgraph problems*

Subgraph problems can be divided into subgraph descriptive heterogeneity problems on one side and subgraph no textual description problems on the other (see Fig. 10). We make a distinction between the heterogeneity of description and the absence of description of certain characteristics of the entity.

For LPT 4.1, an example would be a pair of datasets describing a resource with a different amount of information (e.g., the city of New York with its name, population and geographic location on one side and its name and location on the other).

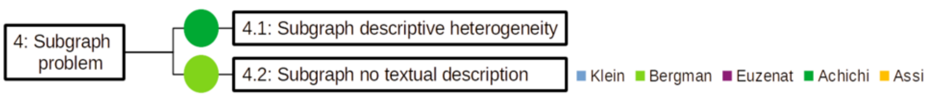


Figure 10. Part of the classification of LPTs: Subgraph problems.

5.1.5. *Graph problems*

Graph problems can be divided into eight levels (see Fig. 11). Graph problems level had to be explicitly added to the classification, because, we distinguish more general problems from those presented at the RDF triplet level. Problems like scalability and expressiveness of some languages compared to others concerning for example the expression of negation.

For LPT 5.4.4, an example would be a pair of datasets that use distinct languages with differences in the representation of logical notions (e.g., a language that directly expresses class disjunctions (A disjoins B) on the one hand and a language requiring the use of negation (A subclass-of (NOT B) on the other). For LPT 5.4.6, an example would be a pair of datasets describing the population and dynamics of the same city but at different times. For LPT 5.6, an example would be a pair of datasets where at the level of the graphs the description patterns would be identical whereas they would be different descriptions (e.g., two sets of triples composed only of individuals of the class person but with on one side a single reflexive relation ”hasSister” and on the other a single reflexive relation ”hasBrother”). This type of problem will be especially useful for the embedding graph.

Once again, we note some differences in the distribution of each problems by level as done by the authors, such as Achichi, who only reports heterogeneities related to the heterogeneity of graph conceptual keys and the timeliness of graph conceptual datasets.

We added the LPT 5.8, as we think it could be useful in the development of hybrid techniques mixing IM and OM where the absence of TBox would be perceived as a problem.

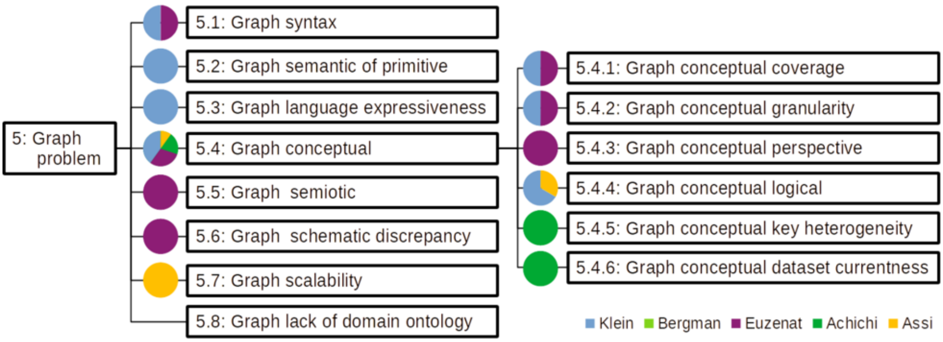


Figure 11. Part of the classification of LPTs: Graph problems.

5.2. *Formalization of the LPT classification*

We formalized the LPT classification, as illustrated in Figure 12. We use the `lpt` prefix as the namespace for our classification. The `lpt:LPT` class is the primary class in our model. To maximize reuse, we rely on established vocabularies as much as possible instead of creating new classes and properties. Especially, we used SKOS, RDF-S, Dublin Core and PROV-O [27] to describe the classes. Additionally, we used the DCAT vocabulary [28] to define datasets and their distributions. We introduced the class `lpt:PairOfDatasets` to represent a couple of `dcat:Datasets` that is or need to be linked. This class reifers the pair into an object that can be described on its own e.g., status of linking, date of linking, source of linking; we do not describe these here. The `lpt:occursIn` property is the key relation in our model: it encodes the fact that a certain linking problem type occurs/appears in a certain pair of datasets. To provide a detailed description of such a pair, instances of the `lpt:PairOfDatasets` class are linked to two separate individuals of the `dcat:Dataset` class using the `lpt:hasSource` and `lpt:hasTarget` properties. The property `prov:wasInfluencedBy` connects the `lpt:LPT` class to the underlying heterogeneities reported by various authors and encoded with the class `lpt:Heterogeneity`. The bibliographic sources from which we derived the descriptions of the heterogeneities that guided our classification are captured with the property `prov:wasDerivedFrom` to an object in the BIBO ontology [29].

An example of instantiation of the LPT classification model is provided in Figure 13. This example is in fact a real example of the appearance of the LPT 2.1.1 problem called `Predicate terminological synonymy` in the `lpt:datasetOAEI101` and `lpt:datasetOAEI205` datasets accessible from `https://oaei.ontologymatching.org/tests/101/onto.rdf` and `https://oaei.ontologymatching.org/tests/205/onto.rdf` respectively. This pair of RDF datasets is made available by OAEI to allow future participants to test their tools.

This classification is currently being made available on the web in OWL format. The choice of this representation language was made with a view to encoding a hierarchical classification for future use by software solutions exploiting the inference capabilities

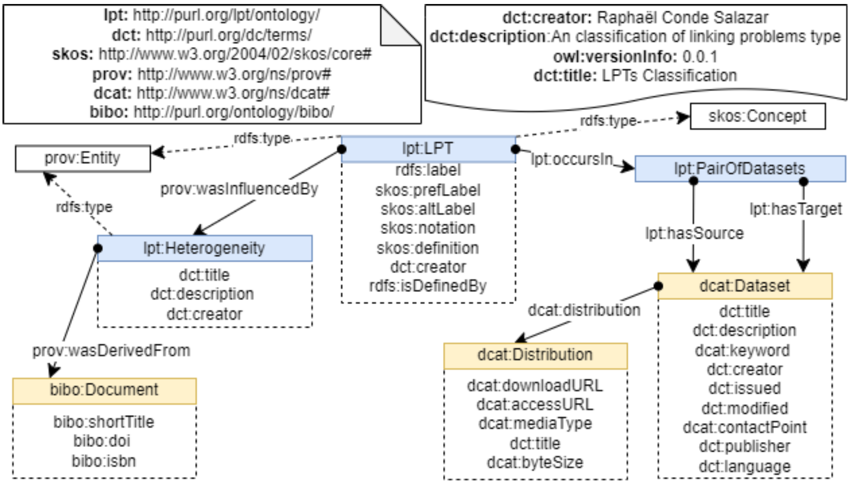


Figure 12. Conceptual model for defining Linking Problem Types.

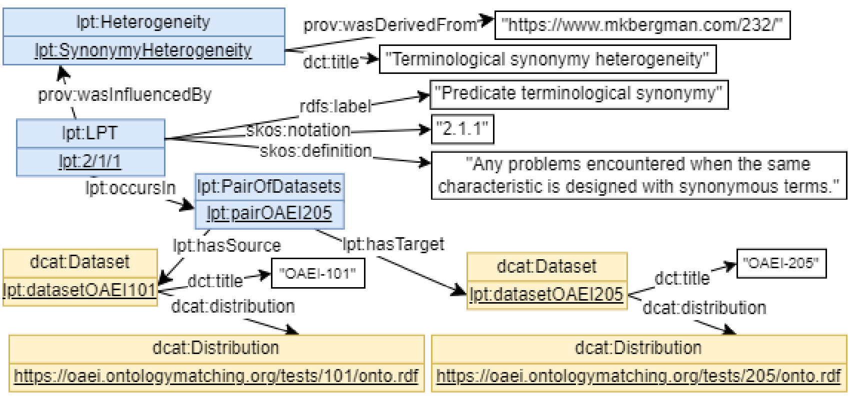


Figure 13. An example of formalization of a LPTs in RDF.

provided by this language. This hierarchisation based on the `rdfs:subClassOf` property involving classes for each LPT (e.g., `LPT_1_1_2 rdfs:subClassOf LPT_1_1`) is not represented here for lack of space.

6. Conclusion

Data linking allows similar entities to be linked, so that semantic data spread over several heterogeneous datasets can be used more effectively. In this paper, we therefore propose a formalized classification of the different types of problems that can be encountered when linking RDF datasets. We hope that this classification will help the data linking community to better identify the problems that may arise when two RDF datasets with heterogeneous terminology, structure, and logic need to be linked. Establishing a precise profile, as close as possible to the RDF data to be processed, should allow a better choice of the algorithmic module(s) needed to solve a data linking task, in an attempt to improve the performance of existing data linking tools, most of which use generic solutions.

In the future, we plan to continue to develop our classification by continuing to provide, for example, for each LPT described, examples of real cases from pairs of datasets used during data linking competitions. The different techniques capable of solving these LPTs will also be attached. Ultimately we want to make an OWL ontology that we will of course make available online for all users in the field of data linking.

This work is achieved in the context of the DACE-DL (Data-Centric AI-driven Data Linking) project [5] which proposes a paradigm shift in data linking by focusing on a bottom-up, data-centric methodology [11]. The objective of this research project is to use machine learning techniques and representation learning models to improve data linking by facilitating the application of the right linking tool to the relevant linking problem. Thus the need to formalize a classification of linking problem types. We therefore envision our classification to be used to determine, via learning processes, the relevant specific linking tool modules necessary for data linking according to the different problems exposed by a pair of datasets in order to provide a more specific solution to a linking task than current approaches.

Another perspective of this work, is to experiment an unsupervised machine learning process to categorize different pairs of datasets for which we would have manually determined the different LPTs potentially exposed. The goal of such experimentation would be to verify our grouping operated via the LPT classification can be corroborated by a categorization performed via a machine learning process. These dataset pairs are taken from various datalinking benchmarks such as OAEI. Each pair is documented with the different LPTs they expose, an additional file containing the different alignments that should theoretically be obtained after running a linking tool and the linking tool that has been tested as the best performing.. Other information on these datasets is provided (i.e. description, year of creation, origin, type of alignment(T-Box/Schema matching, Instance matching or link discovery, Instance and schema matching and Tabular data to Knowledge Graph matching). In our next project, we aim to set up an automated software solution that would receive as input a pair of datasets that we are trying to link and as output the LPTs that they expose.

Acknowledgement

This work has been supported by the DAta-CEntric AI-driven Data Linking project (DACE-DL - https://dace-dl.github.io/ - ANR-21-CE23-0019).

References

[1] Berners-Lee T, Hendler J, Lassila O. The semantic web. Scientific american. 2001;284(5):34-43.

[2] Beek W, Raad J, Wielemaker J, van Harmelen F, editors. sameas. cc: The closure of 500m owl: sameas statements. Springer; 2018.

[3] Correndo G, Penta A, Gibbins N, Shadbolt N, editors. Statistical analysis of the owl: SameAs network for aligning concepts in the linking open data cloud. Springer; 2012.

[4] Halpin H, Hayes PJ, McCusker JP, McGuinness DL, Thompson HS, editors. When owl: Sameas isn't the same: An analysis of identity in linked data. Springer; 2010.

[5] Ding L, Shinavier J, Shangguan Z, McGuinness DL, editors. SameAs networks and beyond: Analyzing deployment status and implications of owl: sameAs in linked data. Springer; 2010.

[5] Data-Centric AI-driven Data Linking. https://dace-dl.github.io/

[6] Nentwig M, Hartung M, Ngonga Ngomo AC, Rahm E. A survey of current link discovery frameworks. Semantic Web. 2017;8(3):419-36.

[7] Euzenat J, Meilicke C, Stuckenschmidt H, Shvaiko P, Trojahn C. Ontology alignment evaluation initiative: Six years of experience. In: Journal on data semantics XV. Springer; 2011. p. 158-92.

[8] Algergawy A, Faria D, Ferrara A, Fundulaki I, Harrow I, Hertling S, et al., editors. Results of the ontology alignment evaluation initiative 2019. vol. 2536; 2019.

[9] Abd Nikooie Pour M, Algergawy A, Amini R, Faria D, Fundulaki I, Harrow I, et al., editors. Results of the ontology alignment evaluation initiative 2020. vol. 2788. RWTH; 2020.

[10] Pour M, Algergawy A, Amardeilh F, Amini R, Fallatah O, Faria D, et al., editors. Results of the ontology alignment evaluation initiative 2021. vol. 3063. CEUR; 2021.

[11] Todorov K, editor. Datasets First! A Bottom-up Data Linking Paradigm; 2019.

[12] Euzenat J. Extraction de clés de liage de données (résumé étendu). In: 16e conférence internationale francophone sur extraction et gestion des connaissances (EGC). Hermann; 2016. p. 9-12.

[13] Atencia M, David J, Scharffe F, editors. Keys and Pseudo-Keys Detection for Web Datasets Cleansing and Interlinking. Springer; 2012.

[14] Symeonidou D, Armant V, Pernelle N, Saïs F, editors. Sakey: Scalable almost key discovery in RDF data. Springer; 2014.

[15] Symeonidou D, Armant V, Pernelle N. BECKEY: Understanding, comparing and discovering keys of different semantics in knowledge bases. Knowledge-Based Systems. 2020;195:105708.

[16] Babic B, Nesic N, Miljkovic Z. A review of automated feature recognition with rule-based pattern recognition. Computers in industry. 2008;59(4):321-37.

[17] Käfer T, Harth A. Rule-based Programming of User Agents for Linked Data. LDOW@ WWW. 2018;2073.

[18] Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering. 2017;29(12):2724-43.

[19] Dai Y, Wang S, Xiong NN, Guo W. A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics. 2020;9(5):750.

[20] Harispe S, Ranwez S, Janaqi S, Montmain J. Semantic similarity from natural language and ontology analysis. Synthesis Lectures on Human Language Technologies. 2015;8(1):1-254.

[21] Klein M, editor. Combining and relating ontologies: An analysis of problems and solutions; 2001.

[22] Bergman M. Sources and classification of semantic heterogeneities. Web Blog: AI3-Adaptive Information, Adaptive Innovation, Adaptive Infrastructure. 2006.

[23] Pluempitiwiriyawej C, Hammer J. A classification scheme for semantic and schematic heterogeneities in XML data sources. TR00-004, University of Florida, Gainesville, FL. 2000.

[24] Euzenat J, Shvaiko P. Ontology matching. vol. 18. Springer; 2007.

[25] Achichi M, Bellahsene Z, Ellefi MB, Todorov K. Linking and disambiguating entities across heterogeneous RDF graphs. Journal of Web Semantics. 2019;55:108-21.

[26] Assi A, Mcheick H, Dhifli W. Data linking over RDF knowledge graphs A survey. Concurrency and Computation: Practice and Experience. 2020;32(19):e5746.

[27] Lebo T, Sahoo S, McGuinness D, Belhajjame K, Cheney J, Corsar D, et al. Prov-o: The prov ontology. W3C recommendation. 2013;30.

[28] Albertoni R, Browning D, Cox S, Beltran AG, Perego A, Winstanley P. Data catalog vocabulary (DCAT)-version 2. World Wide Web Consortium. 2020.

[29] D'Arcus B, Giasson F. Bibliographic ontology specification. Madrid: Biblioteca Nacional Española. 2009.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230015

Semantics for Implementing Data Reuse and Altruism Under EU's Data Governance Act

Beatriz ESTEVES [a], Víctor RODRÍGUEZ DONCEL [a]
Harshvardhan J. PANDIT [b] Dave LEWIS [c]
[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Spain*
[b] *ADAPT Centre, Dublin City University, Ireland*
[c] *ADAPT Centre, Trinity College Dublin, Ireland*

Abstract. *Purpose:* Following the impact of the GDPR on the regulation of the use of personal data of European citizens, the European Commission is now focused on implementing a common data strategy to promote the (re)use and sharing of data between citizens, companies and governments while maintaining it under the control of the entities that generated it. In this context, the Data Governance Act (DGA) emphasizes the altruistic reuse of data and the emergence of data intermediaries as trusted entities that do not have an interest in analysing the data itself and act only as enablers of the sharing of data between data holders and data users. *Methodology:* In order to address DGA's new requirements, this work investigates how to apply existing Semantic Web vocabularies to (1) generate machine-readable policies for the reuse of public data, (2) specify data altruism consent terms and (3) create uniform registers of data altruism organisations and intermediation services' providers. *Findings:* In addition to promoting machine-readability and interoperability, the use of the identified semantic vocabularies eases the modelling of data-sharing policies and consent forms across different use cases and provides a common semantic model to keep a public register of data intermediaries and altruism organisations, as well as records of their activities. Since these vocabularies are openly accessible and easily extendable, the modelling of new terms that cater to DGA-specific requirements is also facilitated. *Value:* The main results are an ad-hoc vocabulary with the new terms and examples of usage, which are available at *https://w3id.org/dgaterms*. In future research, this work can be used to automate the generation of documentation for the new DGA data-sharing entities and be extended to deal with requirements from other data-related regulations.

Keywords. Data Governance Act, Semantic Web, Machine-readable policies, Data intermediaries, Data altruism, Registers of Activities

1. Introduction

In February 2020, following the impact of the General Data Protection Regulation (GDPR) [1] in the specification of new data subject rights and in the implementation of new obligations on the entities processing personal data, the European Commission published a document establishing its *strategy for data*, including a package of new regulation proposals to legislate the usage of non-personal and public data, the activity of digital services and digital markets and the development of *common European data spaces* [2]. While putting the data in the centre of this transformation, by making it available to all and facilitating its flow between sectors, the interests of data subjects and data holders will be kept by having clear data-sharing policies to govern the usage and access to data and trusted entities that enable said sharing while enforcing compliance with the new regulations.

In particular, the Regulation of the European Parliament and of the Council on European data governance, the Data Governance Act (DGA), was proposed to improve the availability of public data, promote trust in data intermediation service providers and data altruism organisations as enablers of data-sharing between data holders and data users for purposes of general interest, and to establish a new supra-national authority charged with overseeing the activities of such entities, the European Data Innovation Board [3]. The DGA, along with the proposed visions for an European Health Data Space [4] and the Data Act [5], put an emphasis on the altruistic reuse of data – in the Health Data Spaces case to address the challenges of access and sharing to electronic health data – currently trapped within various institutions in the EU and unavailable to be used by all. The key challenges to be addressed to realise these visions are related to the:

- Ch1. *Availability / Discovery of datasets*: without the promotion and technical support for the development of common data spaces and trusted data sharing entities, data subjects and data holders will not have automated tools to share their data to be reused for common good purposes, nor solutions to support them in the exercising of their rights, and data users will not have tools to search for the data they seek.
- Ch2. *Establishment of conditions for usage and access to data*: without standards and metadata vocabularies to express interoperable, machine-readable policies, the establishment of conditions for usage and access to personal, non-personal and public-sector data, based not only on legal but also on ethical, organisational and social norms, will provoke interoperability issues between entities providing and seeking access to data.
- Ch3. *Production of Documentation*: without keeping records of their activities in a structured format, data intermediation service providers and data altruism organisations will rely on manual processes to produce documentation that demonstrates their accountable and responsible practices.

Therefore, Semantic Web vocabularies, such as the W3C's Data Privacy Vocabulary (DPV)[1], Open Digital Rights Language (ODRL)[2] or Data Catalog Vocabulary (DCAT)[3], have an important role to play in these processes as they are interoperable and form common standards that enable machine-readable tools to be used for the automation of tasks. DCAT and ODRL are W3C Recommendations to describe data catalogues published on the Web and to express usage rules

[1] https://w3id.org/dpv
[2] https://www.w3.org/TR/odrl-model/, https://www.w3.org/TR/odrl-vocab/
[3] https://www.w3.org/TR/vocab-dcat-2/

over digital datasets, respectively. DPV is a W3C Community Group Report, which has recently published a stable version 1, aimed at providing a complete, open-access set of taxonomies to express machine-readable metadata about the use and processing of personal data, such as taxonomies for legal entities, purposes, types of processing activities, legal basis, types of data, rights or technical and organisational measures. By combining the usage of these standards and specifications to automate the discovery of datasets (Ch1), specify policies for the reuse and sharing of data (Ch2), and comply with legal obligations (Ch3), such as sending notifications to the competent authorities under the DGA, this work will enable organisations to gradually move from completely manual processes to ones based on utilising automation and technologies to assist in ensuring correctness and scalable architectures on the data-sharing services ecosystem.

In order to address the identified challenges, we determined what reuse conditions are necessary to specify *how* to share data, *who* are the new involved stakeholders, and *what* documents are required to comply with the new law. Therefore, the following research objectives are presented as the basis of this work:

RO1. Identify the stakeholders, information items and information flows relevant for the sharing of data compliant with the DGA.
RO2. Identify terms missing from W3C's specifications for representing data-sharing policies and consent terms.
RO3. Generate registers of altruistic and data intermediary activities which can be queried by the competent authorities.

Moreover, the principal contributions of this paper are summarised as follows:

C1. Identification of DGA entities and how data flows between them.
C2. Identification of Use Cases where the usage of semantic vocabularies will assist in the automation of tasks.
C3. Development of ad-hoc vocabulary for representing data-sharing policies, consent and permission terms and registries of activities.
C4. Demonstration of representation of DGA-related information using the mentioned semantic web technologies and the developed vocabulary.

This paper is organized as follows: Section 2 describes the entities and flows of data between entities defined by the DGA in which the usage of semantic technologies can promote the automation of tasks, while Section 3 discusses the state of the art in semantic models for data governance. Section 4 provides an identification of vocabulary terms that can be reused for the purposes of providing examples for policies for the reuse and sharing of data, querying registries of activities and specifying data altruism form terms. Section 5 discusses the impact of our approach on compliance with DGA and its limitations and the last section presents conclusions and future lines of work.

2. Information Flows in the DGA

As the DGA promotes the availability and regulates the sharing of data, a set of information flows, related to the information that needs to be exchanged between data-sharing entities, can be described. In this context, an information flow specifies the information that has to be transmitted from one entity to another or that needs to be kept in a document, such as a record of activity or a public register, to fulfil a certain DGA requirement. Figure 1 displays a diagram of the identified entities and information flows.

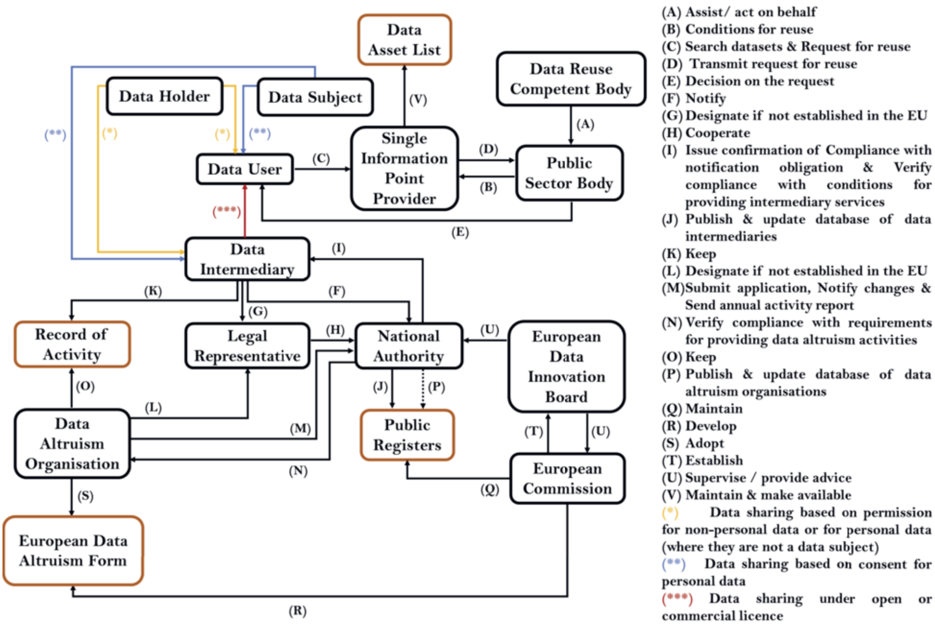


Figure 1. Flows of information between DGA-specified stakeholders. The concepts surrounded by a black box represent (legal) entities and the ones surrounded by an orange box represent newly introduced documents, to be created and maintained by the identified entities. The direction of the arrows represents the direction of the information flow between entities. A simple description of each information flow is provided on the right side of the Figure.

The identified entities can be classified as a data holder, data subject, data user, data intermediation service provider – or data intermediary –, data altruism organisation, legal representative, national, or competent, authority, single information point provider, public sector body, or competent body. Information flows including the soon-to-be-created European Data Innovation Board and European Commission are also displayed in Figure 1. Their definitions are presented below:

Data Subject Individual whose personal data is being processed

Data Holder An entity who has the right to grant access to or to share certain personal data or non-personal data

Data User An entity who has the right to use personal or non-personal data for commercial or non-commercial purposes

Data Intermediation Service Provider An entity who establishes commercial relationships for the data sharing between data subjects and data holders on the one hand and data users on the other

Data Altruism Organisation An non-profit organisation that collects and shares data for altruistic purposes

Public Sector Body An entity or association of entities governed by public law formed by one or more State, regional or local authorities

Legal Representative A representative of a legal entity designated to act on behalf of a data intermediation service provider or altruism organisation

Competent Body An entity designated by a public sector body to provide legal and technical support on the access and reuse of public sector data

Single Information Point Provider An entity who is responsible for receiving and transmitting requests for the re-use of public data

Competent Authorities Authorities in charge of supervising the activity of data intermediation service providers and data altruism organisations and maintaining a public register of said entities

European Data Innovation Board An authority tasked with overseeing the activities of data intermediaries and data altruism organisations

For instance, the data intermediary shares the conditions for data access under an open or commercial licence with the data user (flow represented in Figure 1 with a red arrow) and keeps a record of its activities (flow represented in Figure 1 with the (K) arrow). This diagram is derived from an analysis of Chapters II ('Re-use of certain categories of protected data held by public sector bodies'), III ('Requirements applicable to data intermediation services'), IV ('Data altruism') and VI ('European Data Innovation Board') of the DGA. Each article in these chapters was manually studied to search for interactions between the identified entities and, when a flow of information was identified between more than one entity, the respective interaction was recorded in the diagram. In addition, requirements related to the production of compliance documentation are also recorded in the diagram as they require the recording of information which can be automated with the usage of semantic technologies.

In the next three subsections, we focus on the information flows related to the conditions for the re-use of public data (subsection 2.1), with keeping registers of altruistic and intermediary activities (subsection 2.2), and with data altruism activities (subsection 2.3), where the usage of semantic technologies can best assist the involved entities in automating their flow-related tasks. For each example use case, a methodical study of the involved information flows, and respective items of information that need to be exchanged, was manually performed for each identified flow and systematised in the following subsections.

2.1. Use Case U1: Conditions for the Reuse of Public Data

DGA's Chapter II is dedicated to the 'Re-use of certain categories of protected data held by public sector bodies', including the specification of what categories of data it refers to (Article 3), the information conditions that public sector bodies need to document in order to provide such services (Article 5 and 6) and the description of single information point providers and how they are used by data users to search for and request datasets for re-use (Article 8 and 9). The information that public sector bodies need to provide, and a list of the DGA's source articles where it is mentioned, is available in Table 1. This information, which can be specified with the assistance of a competent body (as represented by the (A) arrow in Figure 1), needs to be shared with the single information point provider (as represented by the (B) arrow in Figure 1), so that data users can search datasets ((C) arrow) and send a request for re-use of data through the single information point ((D) arrow). Single information point providers must also maintain and make available a data asset list (represented by the (V) arrow), including information on available resources and the conditions for their re-use.

Table 1. Information items about public sector bodies' services.

Article	Information items
2.9	Data user/categories of users
5.1	Public sector body information
5.1	Competent body information
5.2	Categories of data
5.2	Purposes for usage and access
5.2, 5.3(a)	Nature of data
5.3(b), 5.3(c)	Processing environment
5.5	Measures to prevent re-identification of data holders/subjects
5.9	Third party recipients
6.2	Fees
8.2	Data format
8.2	Data size
9	Procedure to request reuse

2.2. Use Case U2: Registers of Altruistic and Intermediation Activities

DGA's Chapter III and IV is dedicated to the requirements applicable to entities who wish to provide data intermediation or data altruism services. As for the former, and as is defined in DGA's Article 11, the entities who wish to provide data intermediation as a service need to notify their competent national authority of said intentions (as represented by the (F) arrow in Figure 1), which in turn must publish and maintain an updated public register of intermediaries (as represented by the (J) arrow). The conditions required to perform such service are depicted in Article 12, such as the requirement to appoint a legal representative if the data intermediation entity is not established in the EU (as represented by the (G) arrow), to provide information about the commercial terms of the service, including pricing, date and time of the creation of the data and its geolocation, the format of the data and which formats it can be converted, and about the tools and measures used by the intermediary to facilitate the exchange of data, to protect and ensure its interoperability, and to ease the exercising of data holders and data subjects' rights, including the tools to obtain and withdraw permissions and consent, respectively. In addition to these conditions, the data intermediation service provider must also keep a log record of its activities (as represented by the (K) arrow), which in addition to the previously mentioned conditions, must also contain the entity-related information which is made available in the public register of intermediation providers, including name, public website, legal status, form, ownership structure, subsidiaries, registration number and address of the provider, as well as information regarding the type of provided service.

As for the requirements to open activity as a data altruism organisation, and as is defined in DGA's Article 19, the entities who wish to provide data altruism as a service need to submit an application to their competent national authority (can be the same authority as the one who regulates the national data intermediation service providers) of said intentions (as represented by the (M) arrow in Figure 1). If approved, the national authority must include information about the organisation on a public register of data altruism organisations (as represented by the (P) arrow). Such register includes information regarding the name, public website, legal status, form, registration number of the entity, and the entity's, and its representative if applicable, contact details, as well as information regarding the altruistic purposes behind the activity of the organisation. Moreover, the organisation has to publish and update a uniform and structured record of data altruism activity (as represented by the (O) arrow), which is sent annually to the national authority for verification of compliance (as represented by the (M) and (N) arrows). This record must log the activity of the data altruism organisation and provide information regarding the nature and categories of data that it works with. In addition, such records need to keep logs regarding the users of data, their contact details, the date and duration of the processing, the altruistic purpose for which the data was used, the fees paid by data users or any other sources of income, the technical means used for the processing, as well as a summary of the results of said processing.

2.3. Use Case U3: Allowing Data Altruism

DGA's Article 25 discusses the implementation and development of a "European data altruism consent form", which shall be developed by the European Commission (as represented by the (R) arrow in Figure 1), after consulting with GDPR's watchdog European Data Protection Board, with the soon-to-be-created European Data Innovation Board and with other interested stakeholders. This form should be adopted by the data altruism organisations (as represented by the (S) arrow) to record both the consent given by data subjects to share their personal data and the permissions given by data holders to share their non-personal data. These forms should be kept in both a human and machine-readable format and, as such, are the focus of Use Case U3 as semantic technologies, such as ODRL and DPV, can be used to create an electronic rendition of these documents.

3. Related Work

Jurisdictional data-related laws, such as the DGA or the Data Act, specify obligations and requirements based on the context, purpose, and entities involved in how the data is obtained, used, and shared. For a system to conduct, document, and verify compliance-related activities, such as the maintenance of public registers and records of activities, different types of information need to be represented: (i) the obligations and requirements; (ii) the personal, non-personal and public data and (iii) the data use. Previous work has been performed and published within the general fields of 'regulation compliance' and 'legal metadata expression using vocabularies' [6], to specify how jurisdictional laws can be translated into semantic models for data governance. In the context of this work, we focus on the existing research and solutions, limited to addressing the requirements presented in the Use Cases specified in Section 2, and present the state of the art across the areas of (i) vocabularies to express policies and data activities-related metadata, and (ii) vocabularies to specify information about (personal) data and metadata processing, further described in the next two subsections.

3.1. Vocabularies to Express Policies and Metadata

A recent survey [7] has been published where a set of vocabularies and policy languages are analysed in terms of their capacity to represent the information required to comply with the obligations and rights of GDPR-related entities. In particular, this survey concludes that ODRL is a mature resource, ready to be used for representing policies related to data protection law requirements, which is open source, supported by good documentation and continues to be developed and maintained by a W3C Community Group. The ODRL Information Model [8] is a W3C Standard Recommendation that allows the expression of flexible and complex digital policies, including the possibility to represent permissions and prohibitions to perform certain actions over assets and further restrict said policies using constraints and duties. ODRL also supports the development of extensions, the so-called ODRL profiles[4], that provide a way to add further terms for specific domains which are not present in the core ODRL vocabulary. Though other solutions, such as XACML [9] or LegalRuleML [10], provide a richer expressivity and formal semantics to utilise such resources, ODRL has a convenient extension mechanism and has been proven to work as a policy language to deal with GDPR requirements [11,12]. Other general vocabularies, such as the W3C DCAT Recommendation [13] or the DCMI Metadata Terms (DCT) specification [14], will also be used as they provide terms to describe metadata related with datasets including information about the entities who create and maintain data or temporal and spatial assertions regarding the usage and access to data.

3.2. Legal Vocabularies to Specify Data and its Processing

A vocabulary specifying legal concepts is required for expressing policies aligned with data-related regulations and, in the case of this work, one that can easily complement and be integrated with ODRL, and the other previously mentioned vocabularies, to express examples related to the Use Cases specified in Section 2. While no work within the state of the art provides concepts to deal with DGA requirements, several vocabularies have been developed to cover GDPR concepts that can be reused. In particular, and as confirmed by the previously cited survey on data protection vocabularies [7], DPV's [15] set of taxonomies provides the most complete set of vocabularies to express information regarding data, entities, processing activities, purposes, legal basis, rights, risks and consequences, tech-

[4] ODRL Profile Best Practices - `https://w3c.github.io/odrl/profile-bp/`

nical and organisational measures, rules or technologies. Moreover, there already exists published work that uses DPV to create a semantic model for the representation of information related to GDPR's Register of Processing Activities [16]. As such, DPV will be the base vocabulary upon which this work will be developed.

4. Extending W3C vocabularies to cover DGA requirements

As covered by the previous sections, there is a gap in the representation of information brought by the DGA requirements, in particular, to specify conditions for the reuse of public data (further developed in Section 4.1), to populate public registers of data intermediation service providers and data altruism organisations and record their activities (further developed in Section 4.2) and to create a common data altruism form for data subjects' consent and data holders' permissions (further developed in Section 4.3). In the following subsections, we discuss terms of existing standards and specifications that can be used to represent some of the information items described in Section 2 and define the terms that are missing in an open-source ad-hoc vocabulary, to cover the identified Use Cases. In addition, for each Use Case, we also provide examples to demonstrate their applicability.

4.1. Policies for the Reuse and Sharing of Public Data

As described in Section 2.1, public sector bodies need to provide single information point providers information regarding the data resources they own and the conditions for their usage, so that these providers can make available and maintain a searchable asset list, which data users can use to search and request datasets for re-use. Table 2 contains the DPV, DCAT and DCT's terms that can be reused to model some of the concepts identified in Table 1.

Table 2. Information items that need to be modelled to express the conditions of re-use of public sector bodies datasets and respective terms from existing vocabularies that can be reused.

Article	Information items	Terms from existing vocabularies
5.1	Public sector body information	`dpv:hasName`, `dpv:hasContact`
5.1	Competent body information	`dpv:hasName`, `dpv:hasContact`
5.2	Categories of data	`dpv:hasData`, `dpv:Data`
5.2	Purposes for usage and access	`dpv:hasPurpose`, `dpv:Purpose`
5.3(a)	Nature of data	`dpv:hasData`, `dpv:AnonymisedData`, `dpv:PseudonymisedData`
5.3(b), 5.3(c)	Processing environment	`dpv:ProcessingContext`, `dpv:hasLocation` `dpv:WithinVirtualEnvironment`, `dpv:WithinPhysicalEnvironment`
5.5	Technical and operational measures to prevent re-identification of data holders/subjects	`dpv:Deidentification`
5.9	Third party recipients	`dpv:ThirdParty`
8.2	Data format	`dcat:mediaType`, `dct:format`
8.2	Data size	`dct:extent`

In addition to these, to specify data users, public sector bodies, competent bodies and single information point providers, we added four new classes of entities (as subclasses of `dpv:LegalEntity`) to our vocabulary to represent these terms, `DataUser`, `PublicSectorBody`, `DataReuseCompetentBody`, and `SingleInformationPointProvider`, respectively. EU, national, regional, local and sectorial-level single information point providers are also modelled as subclasses of `SingleInformationPointProviders`, as depicted in DGA's Article 8. To be able to classify the nature of the data held by public sector bodies, as specified in Article 3.1, we also added four new subclasses of `dpv:Data`, `ConfidentialData`, `CommerciallyConfidentialData`, `StatisticallyConfidentialData` and `IntellectualProperty` to represent data protected through `CommercialConfidentialityAgreements` or through `StatisticalConfidentialityAgreements` and data protected by intellectual property rights.

Moreover, the following legal basis for the transfer of public sector body-held data, as specified in Article 5, are also included in our vocabulary, as subclasses of `dpv:DataTransferLegalBasis`: `A5-9` for permissions to transfer, `A5-11` for model contractual clauses, and `A5-12` for adequacy decisions. `DataReusePolicy`, `DataTransferNotice` and `ThirdCountryDataRequestNotice` concepts were also added, as subclasses of DPV's policy and notice concepts, to represent the conditions for reuse of data and the notice provided to the owners of said data. As there were no concepts identified to model the searchable asset list maintained by the `SingleInformationPointProviders` and the procedure to request datasets, both concepts were modelled as `DataAssetList` and as `DataReuseRequestProcedure` and as subclasses of `dpv:OrganisationalMeasure`.

To showcase the usage of existing and newly created terms, an example `DataReusePolicy` for reusing the *http://example.com/dataset_001* dataset, that can be used until the end of 2023 for the purpose of `ScientificResearch`, can be found in Listing 1[5]. It is modelled as an ODRL offer as it proposes the terms of usage of the dataset, but does not grant any privileges to the data user. Said policy can be used by single information point providers to maintain an updated list of available assets and the conditions for their usage. Listing 2 provides an example of a `DataAssetList` published by a `SingleInformationPointProvider`, using the existing and the newly created terms. This list contains the previously mentioned dataset, *http://example.com/dataset_001*, with additional metadata regarding the category of data it contains, the policy that governs its usage, *http://example.com/policy_001*, the format and size of the data and the fees charged by the publisher of the dataset.

4.2. Querying public registers of data intermediaries

As described in Section 2.2, data intermediation service providers and data altruism organisations need to submit information about their activity to a public register of such entities in order to have a centralised database of entities, which can be used by data users, data holders or data subjects to retrieve or publish data, for instance for altruistic purposes.

[5]The prefixes and namespaces described in Listing 1 are valid for all Listings.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX odrl: <http://www.w3.org/ns/odrl/2/>
PREFIX dpv: <https://w3id.org/dpv#>
PREFIX dpv-pd: <https://w3id.org/dpv/dpv-pd#>
PREFIX dpv-gdpr: <https://w3id.org/dpv/dpv-gdpr#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ex: <http://example.com/>
PREFIX : <http://anon/dgaterms#>

ex:policy_001 a odrl:Offer, :DataReusePolicy ;
    odrl:permission [
        odrl:target ex:dataset_001 ; odrl:action :Reuse ;
        odrl:assigner ex:publicsectorbodyX ;
        odrl:constraint [
            odrl:and [
                odrl:leftOperand odrl:dateTime ;
                odrl:operator odrl:lteq ;
                odrl:rightOperand "2023-12-31"^^xsd:date ], [
                odrl:leftOperand odrl:purpose ;
                odrl:operator odrl:isA ;
                odrl:rightOperand :ScientificResearch ] ] ] .
ex:publicsectorbodyX a :PublicSectorBody ;
    dpv:hasName "Public Sector Body X" ;
    dpv:hasContact "mailto:publicsectorbodyX@email.com" ;
    :hasCompetentBody [
        a :DataReuseCompetentBody ; dpv:hasName "Competent Body X" ;
        dpv:hasContact "mailto:competentbodyX@email.com" ] .
```

Listing 1: ODRL Offer policy set by the Public Sector Body X that permits the re-use of a dataset until the end of 2023 for scientific research.

```
ex:SIPPA_assets a :DataAssetList, dcat:Catalog ;
    dct:description "Asset list maintained by SIPPA" ;
    dct:created "2022-12-10"^^xsd:date ;
    dct:publisher ex:SIPPA ; dcat:dataset ex:dataset_001 .
ex:SIPPA a :SingleInformationPointProvider .
ex:dataset_001 a dcat:Dataset ; dct:publisher ex:publicsectorbodyX ;
    dpv:hasData :StatisticallyConfidentialData ;
    dct:description "Dataset with statistically confidential data" ;
    dct:created "2022-12-04"^^xsd:date ;
    odrl:hasPolicy ex:policy_001 ; :hasFee "0€"^^xsd:string ;
    dcat:mediaType <iana.org/assignments/media-types/text/csv> ;
    dct:extent "5.6MB"^^xsd:string .
```

Listing 2: Data asset list maintained by the Single Information Point Provider A.

```
ex:publicregistry_DI_PT a :RegisterOfDataIntermediationServiceProviders ;
    dct:description "Public register of intermediaries working in PT" ;
    dct:created "2023-12-15"^^xsd:date ;
    dct:modified "2023-12-23"^^xsd:date ;
    dct:publisher ex:nationalauthority_PT ;
    :hasDataIntermediationServiceProvider ex:DISP_Y .
ex:nationalauthority_PT a :DataIntermediationAuthority ;
    dpv:hasName "Data Intermediation Authority of Portugal" ;
    dpv:hasContact "mailto:nationalauthority_PT@email.com" ;
    dpv:hasJurisdiction "PT" .
ex:DISP_Y a :DataCooperative ;
    dpv:hasName "Data Cooperative Y" ; dpv:hasAddress "Lisboa, Portugal" ;
    dct:description "Provider of anonymised geolocation data" ;
    dcat:landingPage <http://cooperativeA.com/> ;
    dct:date "2023-12-23"^^xsd:date .
```

Listing 3: Example of a public register of data intermediation service providers.

Using the existing and the newly created terms, an example of a register of data intermediation service providers can be found in Listing 3. Due to restrictions in the size of this publication, we do not provide an example of a public register of a data altruism organisation, as both types of public registers contain similar information and will have similar semantic representations. The register `ex:publicregistry_DI_PT` will have a complete list of intermediaries operating in Portugal. Beyond the stored metadata regarding the national authority `ex:nationalauthority_PT` and creation dates, the register has already a registered `DataCooperative` company: `ex:DISP_Y`.

DPV's `hasName`, `hasContact` and `hasAddress` and DCAT's `landingPage` can be used to provide information about the providers of data intermediation or data altruism services, while DCT's `description`, `created`, and `publisher`, can be used to describe metadata about the register, including its creation date and its publisher. In addition to these terms that can be reused from existing standards and specifications, to specify a data intermediation service provider (as a subclass of `dpv:LegalEntity`), or one of its types, we added four new classes of entities to our vocabulary to represent these terms, `DataIntermediationServiceProvider`, `DataCooperative`, `DataIntermediationServiceProviderForDataHolder`, and `DataIntermediationServiceProviderForDataSubject`. Moreover, data altruism organisations are modelled as a subclass of `dpv:NonProfitOrganisation`. Information related to the nature of the entity, as specified in Article 11.6(b), to represent the legal status, form, ownership structure, subsidiary and registration number of an entity, is out of the scope of this contribution as it refers to organisational details. However, as a future contribution, upper ontologies such as GIST [17] or Schema.org [18] can be explored, and if necessary extended, to include such concepts.

Furthermore, a `PublicRegister` class was also added to our vocabulary, and its respective subclasses `RegisterOfDataIntermediationServiceProviders` and `RegisterOfDataAltruismOrganisations` to represent public registers of

```
SELECT DISTINCT ?Provider ?Name ?Web WHERE {
    ?Provider a :DataCooperative .
    ?Provider dpv:hasName ?Name .
    ?Provider dcat:landingPage ?Web . }
```

Listing 4: SPARQL query to retrieve data cooperatives.

```
ex:altruism_logs a :RegisterOfDataIAltruismActivity ;
    dct:description "Activity logs of the Data Altruism Organisation A" ;
    dct:created "2023-11-04"^^xsd:date ;
    dct:modified "2023-11-13"^^xsd:date ;
    dct:publisher ex:altruism_A ; dcat:record ex:log_001 .
ex:altruism_A a :DataAltruismOrganisation ;
    dpv:hasName "Data Altruism Organisation A" ;
    dpv:hasAddress "Lisboa, Portugal" ;
    dcat:landingPage <http://example.com/altruism_A> .
ex:log_001 a dcat:CatalogRecord ;
    dct:created "2023-11-13"^^xsd:date ;
    :hasDataUser ex:userZ ; :hasFee "1000€"^^xsd:string ;
    dpv:hasPersonalDataHandling [
        dct:description "Download and reuse anonymised health records to
        ↪  improve healthcare" ;
        dpv:hasProcessing :Download, :Reuse ; dpv:hasDuration 6226453 ;
        dpv:hasPurpose :DataAltruism, :ImproveHealthcare ;
        dpv:hasPersonalData dpv-pd:HealthRecord ;
        dpv:hasTechnicalMeasure dpv:Anonymisation ] .
ex:userZ a :DataUser ; dpv:hasName "Data User Z" ;
    dpv:hasContact "mailto:user_z@email.com" .
```

Listing 5: Example of a register of data altruism activity logs.

data intermediaries and of altruistic organisations, respectively. By having the public register stored in RDF using the identified and developed semantic vocabularies, such register can then be easily queried, using a query language such as SPARQL to automate the retrieval of information regarding data intermediation service providers. An example of a query for data cooperatives is provided in Listing 4, which will return a list of data intermediation service providers that offer the services of data cooperatives, including their names and public websites.

Listing 5 provides an example of a register of data altruism activity, represented through the newly created concept `RegisterOfDataIAltruismActivity`. Activity logs should be associated with the entities using the data and can be recorded using DPV's `hasPersonalDataHandling` to provide information about the processing of data, including its duration, purpose and (personal) data categories.

```
ex:consentForm_001 a :EuropeanDataAltruismConsentForm ;
    dpv:hasIdentifier <http://example.com/consentForm_001> ;
    dpv:hasDataSubject ex:Anne ; dpv:isIndicatedBy ex:Anne ;
    dpv:isIndicatedAtTime "2022-12-14" ;
    dpv:hasPersonalDataHandling [
        dpv:hasPurpose :DataAltruism, :ImproveTransportMobility ;
        dpv:hasLegalBasis dpv-gdpr:A6-1-a ;
        dpv:hasPersonalData dpv-pd:Location ;
        dpv:hasProcessing dpv:Use, dpv:Store ;
        dpv:hasDataController [
            a dpv:DataController, :DataAltruismOrganisation ;
            dpv:hasName "Company A" ] ] .
```

Listing 6: Data altruism form where data subject Anne consents to the usage of their location data for the altruistic purpose of improving mobility.

4.3. Uniform, Machine-readable Data Altruism Form

As already proven by the examples provided in the former two subsections, the identified vocabularies, as well as the one we developed ourselves, can also be used to automate the production of consent forms for data subjects and permission forms for data holders. By relying on such technologies by design, the European data altruism forms will promote interoperability and can be reused throughout the EU. An example of a consent form, using the term, `EuropeanDataAltruismConsentForm`, by a data subject is provided in Listing 6. In this example, we use an altruistic purpose for processing defined in our vocabulary. As such, we define `DataAltruism` as a subclass or `dpv:Purpose` and we specify seven new purposes that can be used in a data altruism setting: `ImproveHealthcare`, `CombatClimateChange`, `ImproveTransportMobility` (used in Listing 6), `ProvideOfficialStatistics`, `ImprovePublicServices`, `ScientificResearch` and `PublicPolicyMaking`. Additional purposes, mentioned throughout the DGA, are also provided in the ad-hoc vocabulary. Similarly, in Listing 7, we provide an example of a permission form of a data holder.

5. Discussion

A complete list of all defined terms is available at *https://w3id.org/dgaterms*, under an open and permissive licence. The analysis of how semantic technologies can be used to operationalise the DGA yields some promising applications, however, a number of hindrances can be identified. Among the advantages, the following ones should be carefully noted:

- Semantic technologies can help forge a common understanding of the provisions in the regulation.
- Machine-readable policies can be effectively represented in RDF, and executed with appropriate reasoners.

```
ex:permissionForm_001 a dpv:Permission ;
    dpv:hasIdentifier <http://example.com/permissionForm_001> ;
    :hasDataHolder ex:dataHolderA ; dpv:isIndicatedBy ex:dataHolderA ;
    dpv:isIndicatedAtTime "2022-12-15" ;
    dpv:hasPersonalDataHandling [
        dpv:hasPurpose :DataAltruism, :ProvideOfficialStatistics ;
        dpv:hasLegalBasis :A2-6 ; dpv:hasData dpv:AnonymisedData ;
        dpv:hasProcessing dpv:Use, dpv:Store ;
        dpv:hasDataController [
            a dpv:DataController, :DataAltruismOrganisation ;
            dpv:hasName "Company A" ] ] .
```

Listing 7: Permission for data altruism where data holder A allows the usage of their anonymised data for the altruistic purpose of providing official statistics.

- Trust technologies certifying a data altruism consent expression provide legal certainty and encourage data reuse, in the very spirit of the DGA.
- Semantic Web technologies excel at publishing policies on the Web, with JSON-LD serializations easily consumed by Web developers. In addition, RDF can effectively act as a bridge between future expression languages.
- Data altruism may be rewarded in non-economic forms, encouraging in turn further data sharing.

However, the following limitations can be identified:

- Most of the conditions declared in the policies will not be able to be automatically enforced and the declarative nature of the policies will hopelessly lead to data misuse.
- The agreement may not be such if no ontology gains hegemonic spread, if it is not sanctioned by a public authority, or if heavy discrepancies prevent reaching a consensus.

6. Conclusions and Future Work

While powerful, the European strategy for data brings many interoperability challenges that need to be surpassed if we are to implement common spaces to share data between individuals, companies and governments. As such, the effort we made in this work on analysing the requirements of the DGA and providing a common semantic model to record the activities of public sector bodies, data intermediation service providers and altruism organisations are a first step towards conquering this interoperability challenge. As proposed, we identified the stakeholders, information and requirements necessary to model the activities of the new data-sharing entities brought by the DGA and provided a semantic vocabulary, and examples of usage of such vocabulary, that can be used to automate the documentation tasks of these new entities. As for future work, the Data Act and Data Spaces proposals should be explored to improve the quality of this work and promote the interoperability envisioned by the common European data spaces.

Acknowledgments

This work was funded partially by the project Knowledge Spaces: Técnicas y herramientas para la gestión de grafos de conocimientos para dar soporte a espacios de datos (Grant PID2020-118274RB-I00, funded by MCIN/AEI/ 10.13039/501100011033). This research has also been supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813497 (PROTECT) and by the ADAPT SFI Research Centre funded by Science Foundation Ireland and co-funded under the European Regional Development Fund (ERDF) through Grant#13/RC/2106_P2.

This is ongoing work which has been contributed to the Data Privacy Vocabularies and Controls Community Group (DPVCG) as a proposal to integrate the DGA within its outputs.

References

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation); 2018. Available from: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[2] Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions - A European strategy for data; 2020. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0066.

[3] Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act); 2022. Available from: http://data.europa.eu/eli/reg/2022/868/oj/eng.

[4] Proposal for a Regulation of the European Parliament and of the Council on the European Health Data Space; 2022. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52022PC0197.

[5] Proposal for a Regulation of the European Parliament and of the Council on harmonised rules on fair access to and use of data (Data Act); 2022. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2022%3A68%3AFIN.

[6] de Oliveira Rodrigues CM, de Freitas FLG, Barreiros EFS, de Azevedo RR, de Almeida Filho AT. Legal ontologies over time: A systematic mapping study. Expert Systems with Applications. 2019;130:12-30.

[7] Esteves B, Rodríguez-Doncel V. Analysis of Ontologies and Policy Languages to Represent Information Flows in GDPR. Semantic Web Journal. 2022.

[8] Iannella R, Villata S. ODRL Information Model 2.2. W3C Recommendation. 2018.

[9] Parducci B, Lockhart H, Rissanen E. eXtensible Access Control Markup Language (XACML) Version 3.0 [OASIS Standard]; 2013. Available from: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[10] Palmirani M, Governatori G, Athan T, Boley H, Paschke A, Wyner A. LegalRuleML Core Specification Version 1.0; 2021. Available from: https://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/os/legalruleml-core-spec-v1.0-os.html.

[11] Esteves B, Pandit HJ, Rodríguez-Doncel V. ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid. In: 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS PW); 2021. p. 298-306. ISSN: 2768-0657.

[12] Agarwal S, Steyskal S, Antunovic F, Kirrane S. Legislative compliance assessment: framework, model and GDPR instantiation. In: Annual Privacy Forum. Springer; 2018. p. 131-49.

[13] Albertoni R, Browning D, Cox S, Beltran AG, Perego A, Winstanley P. Data Catalog Vocabulary (DCAT) - Version 2; 2020. Available from: https://www.w3.org/TR/vocab-dcat-2/.

[14] DCMI Metadata Terms; 2008. Available from: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

[15] Pandit HJ, Polleres A, Bos B, Brennan R, Bruegger B, Ekaputra FJ, et al. Creating a Vocabulary for Data Privacy: The First-Year Report of Data Privacy Vocabularies and Controls Community Group (DPVCG). In: Panetto H, Debruyne C, Hepp M, Lewis D, Ardagna CA, Meersman R, editors. On the Move to Meaningful Internet Systems: OTM 2019 Conferences. vol. 11877. Springer International Publishing; 2019. p. 714-30. Available from: http://link.springer.com/10.1007/978-3-030-33246-4_44.

[16] Ryan P, Brennan R, Pandit HJ. DPCat: Specification for an Interoperable and Machine-Readable Data Processing Catalogue Based on GDPR. Information. 2022;13(5). Available from: https://www.mdpi.com/2078-2489/13/5/244.

[17] gist ontology v12.0.0; 2023. Available from: https://w3id.org/semanticarts/ontology/gistCore.

[18] Schema.org v22.0; 2023. Available from: https://schema.org/.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230016

The FLINT Ontology: An Actor-Based Model of Legal Relations

Jeroen BRETELER [a], Thom VAN GESSEL [a,1], Giulia BIAGIONI [a], Robert VAN DOESBURG [b]

[a] *TNO*

[b] *TNO, University of Amsterdam*

Abstract. Recording and documenting human and AI-driven normative decision-making processes has so far been highly challenging. We focus on the challenge of normative coordination: the process by which stakeholders in a community understand and agree what norms they abide by. Our aim is to develop and formalize the FLINT language, which allows a high-level description of normative systems. FLINT enables legal experts to agree on norms, while also serving as a basis for technical implementation.

Our contribution consists of the development of an ontology for FLINT and its RDF/OWL implementation which we have made openly accessible. We designed the ontology on the basis of competency questions. Additionally, we validated the ontology by modeling example cases and using the ontology's data model in software tooling.

Keywords. Norms, Normative systems, Legal interpretation

1. Introduction

Recording and documenting human and AI-driven normative decision-making processes is highly challenging. We focus on the challenge of normative coordination: the process by which stakeholders in a community understand and agree what norms they abide by. Our aim is to develop and formalize the FLINT language, which allows a high-level description of normative systems [1,2]. FLINT enables legal experts to agree on norms, while also serving as a basis for technical implementation. It does so by focusing on how norms regulate behavior, and therefore takes a dynamic perspective: it represents norms in terms of normative acts and the pre- and postconditions of these acts.

Although this perspective is useful in practice, a formalization of the concepts used in FLINT is lacking. In this paper, we therefore present the FLINT ontology. We start by describing related ontologies for normative representation in Section 2. We will then explain our methodology in Section 3. In Section 4, we showcase the ontology itself, after which we illustrate how to use the ontology in Section 5. We conclude in Section 6.

[1]Corresponding Author: Thom van Gessel, TNO, Anna van Buerenplein 1, The Hague, The Netherlands; E-mail: thom.vangessel@tno.nl.

2. State of the art

In the legal field, numerous vocabularies and ontologies have been developed to effectively structure and organize data relating to norms [3]. Many solutions address a particular domain or scope, such as norms for assets [4], intellectual property rights [5], or privacy and personal data protection [6,7].

Our interest is in the development of domain-neutral ontologies to represent normative knowledge. Moreover, as we shall motivate in our methodology section, we are interested in representing normative systems [8,9]. Consequently, we focus on representing the full normative action space available to agents across all relevant contexts [10]. In our view, this approach subsumes considerations of normative violations and sanctions. Therefore, our work is at some distance from solutions focusing on obligations and prohibitions or evaluating violations of situations rather than actions, such as LKIF [11], LegalRuleML [12] and recent work by Francesconi & Governatori [13].

Arguably the most relevant comparison for our work is to the UFO-L Legal Core Ontology [14,15].[2] UFO-L provides a formal description of legal relations ([17,18]) in the context of the Unified Foundational Ontology (UFO, [19]). Similar to FLINT, UFO-L has a rich representation for classes of power—liability relations, including ways of identifying what classes of actors can occupy which legal positions within the relation (power holder, liability holder). In adopting the concept of institutional acts, both UFO-L and FLINT can be said to be Searlean [10]. However, we see reason to decouple the concepts of legal relation and institutional act. A second difference between the vocabularies is the representation of institutional facts, which are the parts that together characterize the full state space of a normative system. UFO-L adopts the concept of an event to represent the exercise of legal power. While events can thus encode *that* an action has taken place, they do not offer explicit information about the consequences of that action for the normative state. This further implies that the preconditions under which agents are allowed to take actions cannot be modeled in UFO-L by reference to certain states. Instead, the preconditions would have to be represented by disjunctions of sets of events that need to have taken place in order for the preconditions to hold. We believe an explicit model of normative states enhances normative coordination among stakeholders. That is, FLINT enables stakeholders to understand the influence of acts on the state of the normative system. Vice versa, stakeholders can understand how a state of the normative system determines the deontic status of an act.

3. Methodology

We based the functional requirements for the FLINT ontology on the literature in which the concepts of FLINT are introduced informally [1,2] and described them in terms of competency questions (CQs, see [20]). A fragment of our CQs is displayed in Table 1. The CQs have also been used during the validation process of the level of expressiveness of FLINT. The logical consistency of the ontology has been validated by running HermiT 1.4 on sample data.

[2] Some of the concepts and relations described in UFO-L are implemented as a computational artefact for the Service Contract Ontology (SCO) [16].

Table 1. Examples of competency questions for the FLINT ontology.

Concept	Competency questions
Act	Who can perform the action associated with this act? Who can be a recipient of this act? In what situations is it valid to perform this act? What duties does the actor have after performing this act?
Fact	Is this fact atomic? On which facts does this complex fact depend? What can this agent do? What must this agent do?
Duty	Which acts create this duty? What does the duty holder need to do to terminate this duty?

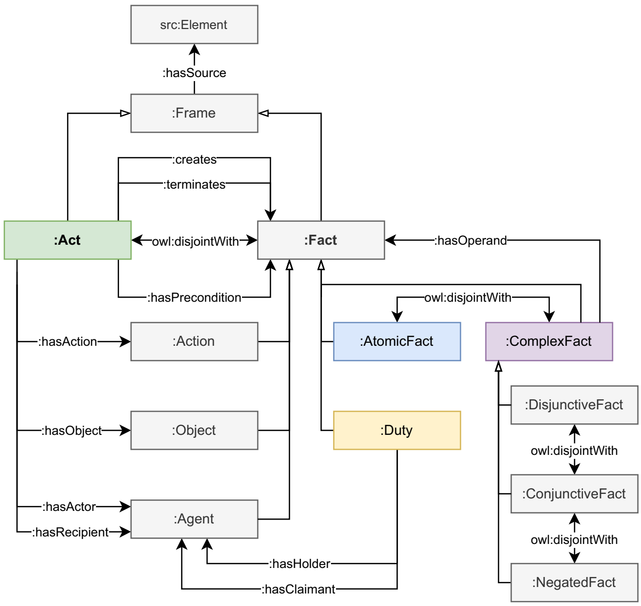


Figure 1. Class diagram of the FLINT ontology. Arrows with white tips represent subclass relations, labelled arrows connect the domain and range of the indicated properties.

4. Ontology overview

The FLINT ontology is a small ontology consisting of only thirteen classes. See Figure 1 for the class diagram. The ontology is implemented in RDF/OWL and is available open source.[3] We describe the main concepts in detail below.

4.1. *Act frames & Fact frames*

The central concept in FLINT is the *frame*: a container that bundles several pieces of information [21]. A FLINT model is made up entirely of these bundles, and the refer-

[3]The repository can be found at https://gitlab.com/normativesystems/knowledge-modeling/flint-ontology. For documentation, see https://normativesystems.gitlab.io/knowledge-modeling/documentation-website/docs/.

ences they make to each other. At the highest level, we make a distinction between two types of frames. *Fact frames* describe matters whose presence or truth value characterizes the state of the normative system. This includes several different kinds of things. First, propositions, which may be complex or atomic and are true or false relative to a state. Second, agents and objects that play a role in the normative system. And third, actions: things an agent can do.

Act frames describe actions that agents might take, which affect the state of the normative system, i.e. the facts. An act frame is connected to fact frames via its properties *hasActor*, *hasRecipient*, *hasObject*, *hasAction*, *hasPrecondition*, *creates* and *terminates*. The first of these three properties describe who can perform the act, who can undergo the act and what objects can be affected by it. In this way, an act frame describes a Hohfeldian power-liability relation between the actor and the recipient. The action related to an act describes what action an agent must take to perform the act. The preconditions of the act describe the circumstances in which the act can be performed legally. Finally, the postconditions of the act are described in terms of the facts that become true and false (using the properties *creates* and *terminates*, respectively) by means of the act.

4.2. Duties

Although a collection of acts with pre- and postconditions can completely describe the valid steps in a process, it is also important to encode what behavior is considered expected according to the norms. This is captured by the *Duty* concept.

We view duties as a special kind of facts, because they are part of the normative state. This means that duties never apply in an absolute manner – like other facts, they are created and terminated by acts. Every duty should have at least one act that creates it (otherwise it never applies) and at least one act that terminates it (otherwise it can never be fulfilled). As a duty represents a Hohfeldian duty-right relation between two parties, it must always have a duty holder and a claimant.

4.3. Classes vs. instances

It is important to note that instances of the classes *Act* and *Fact* are not viewed as concrete acts and facts, but as frames: prototypes of concrete acts and facts as they would be found in a scenario. For example, an act frame can describe applications for a residence permit in general, but not a specific application from an individual. This might raise the question why frames are implemented as individuals rather than as classes like the normative relations in UFO-L. There are three reasons for this.

First, we want to describe intricate relations between act frames and several fact frames without the use of punning. Second, some instantiation is needed not only in a scenario but also in the norms themselves: for example, we want a FLINT model about library regulations to indicate that a borrowed book should be returned. We need a reference to an individual book, not the class of books, to indicate that the borrowed book and the book returned must be one and the same book. Third, while there is a relation between a concrete act in a scenario and an act frame, we believe that conceptually the former is not an instance of the latter.

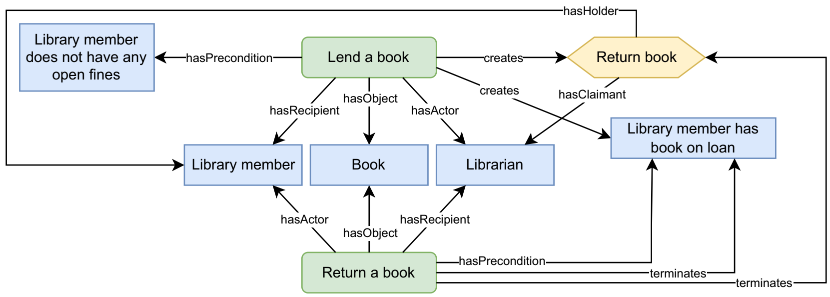


Figure 2. Representation in FLINT of the regulations for lending and returning books. Blue rectangles are instances of *AtomicFact*, green rounded rectangles are instances of *Act*, yellow hexagons are instances of *Duty*. Actions are omitted for readability.

5. Example: library regulations

To illustrate how we intend the FLINT ontology to be used, we describe a toy example of a simple set of library regulations:

1. Books can be loaned to library members if they have no outstanding fines.
2. Library members who borrow books are obligated to return them.
3. Failure to return a book results in a fine.

In our interpretation of these rules, we distinguish three acts: lending a book, returning a book, and giving a fine. The latter act is not found explicitly in the rules, but its existence follows from the assumption that changes to normative states never happen without an act taking place. Therefore, a fine is not viewed as something automatically happening when a book has not been returned, but as the result of an act that has its own pre- and postconditions.[4]

A schematic view of the first two acts and their related facts is shown in Figure 2. This schema shows that a valid performance of the act *Lend a book* creates a duty *Return book*, which can be terminated by a valid performance of the act *Return a book*. That the two operations are related to the same *Book* node as an object, determines that the duty is removed only by returning the same book that was borrowed. Figure 3 shows the act of giving a fine, which has a complex precondition, and terminates the duty to return the book while creating a new duty to pay the fine. Of course, this duty should have a corresponding terminating act in the full picture.

This example illustrates how the FLINT ontology can be used to transform a set of rules into acts and facts, providing insight into what is possible and expected behavior for a particular agent in a given situation. Of course, resolving disputes over what constitutes lawful conduct often requires a much larger graph, in which all parts of laws relevant to the dispute are interpreted as acts and facts.

[4]Note that the rules are not explicit about how much time may elapse before the fine can be given. This could be specified as an additional precondition for the *Give a fine* act.

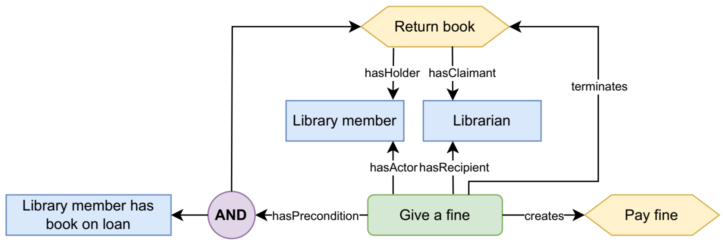


Figure 3. Representation in FLINT of the rules for giving a fine. Purple circles are instances of *ComplexFact*. Nodes related to *Pay fine* are omitted.

6. Conclusion and discussion

We have presented the FLINT ontology and its publicly accessible RDF/OWL implementation. The main aim of the FLINT ontology is to support normative coordination. FLINT does this by describing a normative system using facts to describe a state space, and acts to describe permissible transitions between states. As in [14,15] we thus model legal relations, with a focus on legal powers (in the form of acts) and duties. We found that modeling institutional facts in more detail greatly benefits the clarity of the interpretations of norms. We have also illustrated the intended use of the ontology with a toy example.

There are several research directions connected to FLINT and its ontology in which developments are currently taking place. First, while the FLINT ontology describes interpretations of norms, we are also working towards computational implementations of these interpretations in order to automate normative reasoning. A norm engine based on the FLINT ontology is under development [22], which uses the ontology in combination with SHACL constraints and inference rules to reason about compliance in concrete scenarios.

Second, to support this automated normative reasoning, we are working on a formalization of FLINT as a variant of dynamic logic [23], in order to characterize the decidability and complexity of this logic.

Third, another area of research focuses on the automation of the conversion from legal texts into FLINT frames. This is done by marking some components of FLINT (actions, actors, recipients, and objects) in the legal text through automated semantic role labeling [24,25,26]. The FLINT ontology can be used in this line of work to exchange information between different tools.

Finally, we are working on an extension of the ontology which distinguishes several stages in the transformation from a law in natural language to a FLINT model. This starts with creating annotations in the law text and ends with a full formal model that can be used in conjunction with the norm engine, making all connections between steps explicit so that the formal model is explainable and traceable [27]. With this, we aim to bridge the gap between the theoretical foundations of legal philosophy, and the practical need for normative coordination.

Acknowledgements

We would like to thank the Dutch Ministry of the Interior and Kingdom Relations for financially supporting our research.

References

[1] van Doesburg R, van der Storm T, van Engers T. Calculemus: Towards a formal language for the interpretation of normative systems. In: AI4J workshop at ECAI 2016, The Hague, Netherlands; 2016. p. 73-7.
[2] van Doesburg R, van Engers T. Explicit Interpretation of the Dutch Aliens Act. In: Proceedings of the Workshop on Artificial Intelligence and the Administrative State (AIAS 2019); 2019. p. 27-37.
[3] Leone V. Legal knowledge extraction in the data protection domain based on ontology design patterns. University of Luxembourg, Luxembourg City, Luxembourg; 2021. Available from: http://orbilu.uni.lu/handle/10993/47854.
[4] Iannella R, Villata S, editors. Open Digital Rights Language (ODRL); 2018. https://www.w3.org/TR/odrl-model/.
[5] Abelson H, Adida B, Linksvayer M, Yergler N. ccREL: The Creative Commons Rights Expression Language; 2008. https://www.w3.org/Submission/ccREL/.
[6] Pandit HJ, Fatema K, O'Sullivan D, Lewis D. GDPRtEXT-GDPR as a linked data resource. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15. Springer; 2018. p. 481-95.
[7] Pandit HJ, Polleres A, Bos B, Brennan R, Bruegger B, Ekaputra FJ, et al. Creating a vocabulary for data privacy: The first-year report of data privacy vocabularies and controls community group (DPVCG). In: On the Move to Meaningful Internet Systems: OTM 2019 Conferences: Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, October 21–25, 2019, Proceedings. Springer; 2019. p. 714-30.
[8] Alchourrón CE. Logic of norms and logic of normative propositions. Logique et analyse. 1969;12(47):242-68.
[9] Alchourrón CE, Bulygin E. Normative systems. Springer-Verlag; 1971.
[10] Searle JR. The construction of social reality. Simon and Schuster; 1995.
[11] Hoekstra R, Breuker J, Di Bello M, Boer A. The LKIF Core Ontology of Basic Legal Concepts. In: LOAIT 07: II Workshop on Legal Ontologies and Artificial Intelligence Techniques; 2007. p. 43-63.
[12] Athan T, Governatori G, Palmirani M, Paschke A, Wyner A. LegalRuleML: Design principles and foundations. Reasoning Web Web Logic Rules: 11th International Summer School 2015, Berlin, Germany, July 31-August 4, 2015, Tutorial Lectures 11. 2015:151-88.
[13] Francesconi E, Governatori G. Patterns for legal compliance checking in a decidable framework of linked open data. Artificial Intelligence and Law. 2022:1-20.
[14] Griffo C, Almeida JPA, Guizzardi G. Conceptual modeling of legal relations. In: Conceptual Modeling: 37th International Conference, ER 2018, Xi'an, China, October 22–25, 2018, Proceedings. Springer; 2018. p. 169-83.
[15] Griffo C, Sales TP, Guizzardi G, Almeida JPA. Legal Power-Subjection Relations: Ontological Analysis and Modeling Pattern. In: Conceptual Modeling: 41st International Conference, ER 2022, Hyderabad, India, October 17–20, 2022, Proceedings. Springer; 2022. p. 65-81.
[16] Griffo C, Almeida JPA, Guizzardi G, Nardi JC. Service contract modeling in Enterprise Architecture: An ontology-based approach. Information Systems. 2021;101:101454. Available from: https://www.sciencedirect.com/science/article/pii/S030643791930506X.
[17] Salmond JW. Jurisprudence: Or, The Theory of the Law. Stevens and Haynes; 1907.
[18] Hohfeld WN. Some fundamental legal conceptions as applied in judicial reasoning. The Yale Law Journal. 1913;23(1):16.
[19] Guizzardi G. Ontological foundations for structural conceptual models; 2005.
[20] de Almeida Falbo R. SABiO: Systematic Approach for Building Ontologies. In: Proceedings of the 1st Joint Workshop ONTO.COM/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering; 2014. p. 1-14.

[21] Minsky M. A framework for representing knowledge. MIT, Cambridge; 1974.
[22] Bakker F, Breteler J, van Gessel T, van Engers T, van Doesburg R. A Norm Engine Based on Semantic Web-Compliant Technology; 2023. Presented at AI4Legs 2023: AI for Legislation.
[23] Fischer MJ, Ladner RE. Propositional modal logic of programs. In: Proceedings of the ninth annual ACM symposium on theory of computing; 1977. p. 286-94.
[24] Bakker RM, van Drie RAN, de Boer MHT, van Doesburg R, van Engers T. Semantic Role Labelling for Dutch Law Texts. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference; 2022. p. 448-57.
[25] Bakker RM, de Boer MHT, van Drie RAN, Vos D. Extracting Structured Knowledge from Dutch Legal Texts: A Rule-based Approach. In: Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management; 2022. p. 1-10.
[26] van Drie RAN, Bakker RM, de Boer MHT, Tolios I, Vos D. The Dutch law as a semantic role labeling dataset. In: Nineteenth International Conference on Artificial Intelligence and the Law (ICAIL); 2023. p. 316-22.
[27] Westhofen L, Stierand I, Becker JS, Möhlmann E, Hagemann W. Towards a Congruent Interpretation of Traffic Rules for Automated Driving: Experiences and Challenges. In: Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022) in association with the 35th International Conference on Legal Knowledge and Information Systems (JURIX 2022); 2022. p. 8-21.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230017

Semantifying the PlanQK Platform and Ecosystem for Quantum Applications

Darya MARTYNIUK [a], Naouel KARAM [a], Michael FALKENTHAL [b], Yufan DONG [a] and Adrian PASCHKE [a,c]

[a] *Data Analytics Center DANA, Fraunhofer FOKUS, Berlin, Germany*
[b] *Anaqor AG, Berlin, Germany*
[c] *Institute for Computer Science, Freie Universität Berlin, Germany*

Abstract. Quantum computing is currently experiencing rapid progress. Due to the complexity and continuous growth of knowledge in this field, it is essential to store information in a way that allows an easy access, analysis and navigation over reliable resources. Knowledge graphs (KGs) with machine-readable semantics offer a structural information representation and can enhance the capabilities of knowledge processing and information retrieval. In this paper, we extend the platform and ecosystem for quantum applications (PlanQK) for a KG. Specifically, we describe how the quantum computing knowledge, which is submitted on the platform by researchers and industry actors, is incorporated into the graph. Moreover, we outline the semantic search over the PlanQK KG.

Keywords. Knowledge Graph, Ontology, Semantic Web, Semantic Search, Faceted Search, Ontology-based Search, Quantum Computing

1. Introduction

Quantum computing has experienced notable growth in recent years leading to a corresponding increase of knowledge in this field. Access to a wide spectrum of scientific and business information, including publications, programming code, software, and hardware documentations is without doubt highly beneficial for further research and development of new technologies. However, it also means that finding relevant knowledge in quantum computing related sources has become an increasingly tedious and time-consuming task even when employing search engines. Knowledge graphs (KGs) have grasped significant potential for enhancing capabilities of search systems, facilitating more efficient information retrieval (IR) [1]. Representing data as a graph allows to define formal semantics and improves flexibility for integrating data from heterogeneous sources [1], [2].

In this paper, we introduce an approach for organizing and curating the information available on PlanQK[1] - a collaborative platform and an ecosystem centered on quantum-enhanced applications - in form of a knowledge graph. The mission of PlanQK is to make the research and industrial knowledge about quan-

[1]https://platform.planqk.de/

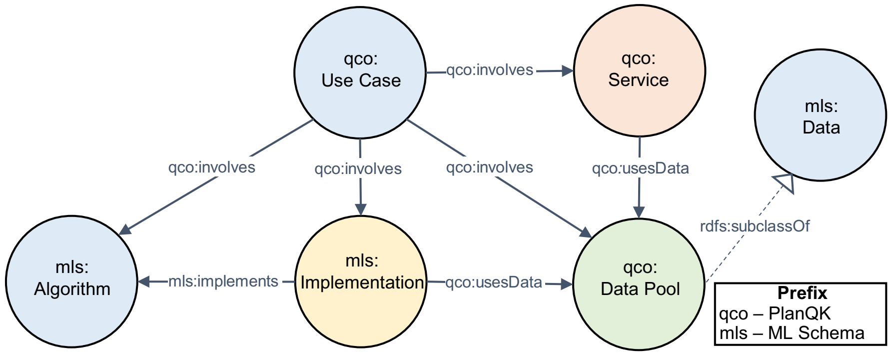


Figure 1. Core knowledge artifacts of the PlanQK ontology.

tum software and solutions easily accessible as well as to offer the capability to deploy, host, execute and monetize quantum services. We present a pipeline for a continuous semantification of the data submitted by user. Furthermore, we outline native semantic search and faceted semantic search, which leverage indexed parts of the PlanQK KG to improve the retrieval of relevant information.

This paper is structured as follows. In Section 2, we briefly introduce the PlanQK KG along with the underlying ontology and provide an overview of the semantification process of the platform data, i.e., its integration into the graph. In Section 3, we describe the realization of KG-based semantic search and semantic faceted search. We give an outlook on future work and a conclusion in Section 4.

2. PlanQK Knowledge Graph

PlanQK KG provides a structured machine-interpretable representation of the platform knowledge by organizing information into named nodes and directed edges that represent specific relations between these nodes[2]. Following the Resource Description Framework (RDF)[3] standard, each node and property is identified by a Uniform Resource Identifier (URI). To query the graph, we utilize SPARQL [3] query language, which is widely used in Semantic Web applications.

The *PlanQK Ontology* [4] serves as a formal schema of the PlanQK KG by defining selected computing concepts and relations between them in the context of the PlanQK platform. The ontology is publicly accessible to the research community including documentation and usage examples[4]. During the creation of the PlanQK ontology, we reviewed existing vocabularies that already encompass definitions of relevant concepts and reused certain ontologies, e.g., ML Schema [5], Subject Resource Application ontology (SRAO) [6], and Software Package Data

[2]Note that the term "knowledge graph" encompasses various definitions in the literature. For an overview and a comprehensive introduction of KGs we refer to [2]. In this paper, we denote a *knowledge graph* as an ontology, which defines domain knowledge along with application data integrated into the ontology, i.e., incorporated with machine-readable semantics.

[3]https://www.w3.org/TR/rdf12-concepts/

[4]https://github.com/PlanQK/semantic-services

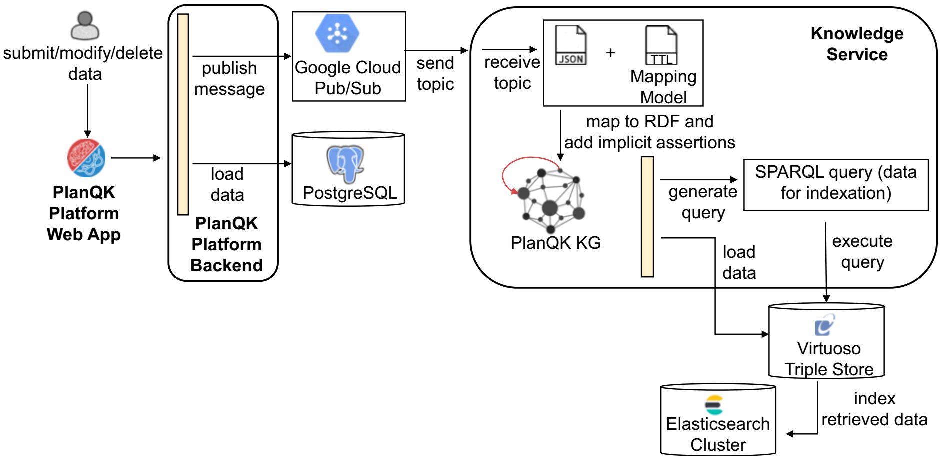


Figure 2. Overview of the platform data integration into the knowledge graph.

Exchange (SPDX) License List [7]. Figure 1 shows the core concepts of the PlanQK ontology, i.e., algorithms, implementations, use cases, data pools and services. Still the primary focus of the semantic annotations in PlanQK lies in describing quantum-related information, it nevertheless encompasses also knowledge related to the classical computing. The reason for this is twofold: Firstly, quantum concepts are often related to classical, e.g., quantum algorithms can be inspired by the idea of their classical counterparts, and, secondly, due to several limitations of the quantum devices in the Noisy Intermediate-Scale Quantum (NISQ) [8] era, quantum-enhanced applications are currently mostly developed in a hybrid manner, i.e., contains both quantum and classical parts [9], [10].

The semantification process of the PlanQK platform is depicted on Figure 2. The frontend of the platform allows users to create, modify and delete knowledge. For example, an user can create a new entry that describes an algorithm with a specific name and computation design, i.e., quantum, classical, or hybrid. The user can further modify the entry by filling various *textual attributes*, e.g., acronym, intent, and solution, and select *annotations* from the controlled vocabulary, e.g., problem class and application area. Once the data is submitted, the backend of the platform manages the information insertion into the relational database[5] and triggers a request to a cloud messaging service, which publishes the message regarding the data changes to the particular channel. To receive notifications from the channel, the knowledge service requires an active subscription. If this is the case, the service obtains a message in the JSON format. The conversion of new data from JSON into RDF is facilitated by Karma [11] integration tool. Karma provides a semi-automatic approach to define a mapping model between

[5]In the context of the PlanQK platform, the decision to use both a relational database and a knowledge graph for data storage is driven by considerations of security and component decoupling. In addition to the explicit content of the platform, the relational database contains the information about user rights. The decoupling of data storage is beneficial to facilitate the future expansion of the knowledge service for an ecosystem of similar platforms.

structured sources and KGs. Given a mapping model, an entity or a collection of similar entities stored in JSON can be dynamically transformed into RDF. However, this mapping is limited to the explicit information conveyed in the received data, i.e., the required inferences must be incorporated into the RDF code prior to its usage for the KG extension. We utilize pre-defined axioms and rules to derive implicit knowledge. An example of such inference is the assignment of a specific type to an algorithm instance, e.g., if the user submits an algorithm and specifies that it solves a classification problem, a relation *rdf:type* to the concept "Classification Algorithm", which is a subclass of "Machine Learning Algorithm", is attached. Finally, the generated RDF is used to update the instance level of the KG. Additionally, the data is indexed and inserted into a search cluster, which is queried during the content search.

We use PostgreSQL[6] as relational database and Google Cloud Pub/Sub[7] as messaging infrastructure. The knowledge service is an extension of the Terminology Service[8] (TS) [12], which was originally created for accessing, developing and reasoning of vocabularies withing the biological and environmental domains. We adjusted and extended the TS implementation to tackle PlanQK requirements, e.g., handling of instance data. The component is developed using Java and Spring Boot framework[9]. The KG is stored in Virtuoso Triple Store[10] and can be queried directly both over a SPARQL endpoint and the REST API of the knowledge service. To store the search index, we use the Elasticsearch (ES) engine[11].

3. Semantic Search

The user of the PlanQK platform has three options to retrieve the knowledge: (i) simple page navigation, (ii) semantic search, and (iii) semantic faceted search. In this section, we describe the last two alternatives.

Traditional keyword-based search approaches that solely relies on queries for literal matching of keywords can not meet the demands of knowledge retrieval since the meaning of the search term and the ambiguous nature of the natural language are not taken into account [13], [14]. The *semantic search* uses the context and the semantics of search terms for improved IR [15]. To generate relevant search results on the PlanQK platform, we index labels and descriptions of instances along with their synonyms, acronyms and broader terms extracted both from the local information, i.e., data submitted by user, and the global information, i.e., data stored in the KG including inferenced knowledge and annotations from the entire PlanQK ontology. Additionally, we employ common techniques for improving the IR such as spelling correction and stop words removal.

Initially, the user submits the search request using the platform frontend. The search term is forwarded to the API of the knowledge service, which is responsible

[6]https://www.postgresql.org/
[7]https://cloud.google.com/pubsub/docs/overview
[8]https://terminologies.gfbio.org/
[9]https://spring.io/projects/spring-boot
[10]https://virtuoso.openlinksw.com/
[11]https://www.elastic.co/de/elasticsearch/

for the retrieval of search results. The service automatically constructs a query that is then performed on an ES cluster. Thereby, the ranking of results is a part of the ES operations. Finally, the results are sent to the backend of the platform, where they undergo filtering depending on user access rights, and are displayed as snippets on the frontend. Note that the information stored in the ES cluster is filled in during the data transformation mentioned in the previous section. This is beneficial because the transformation and the search index in this scenario are generated for individual instances instead of the entire platform data.

The *faceted search* is an intuitive method of IR, where users can explore and refine search results by applying filters, or *facets*, along various dimensions [16], [17]. In our application, we distinguish between two facet types based on their functionality in querying and classifying data: (i) semantic type facets and (ii) property facets. *Semantic type facets* filter instances based on its semantic type. An example is the "problem type" facet for algorithms. The values of the facet are organized as a hierarchical tree with leafs representing the concept "Problem Type" and its subclasses, e.g., 'Machine Learning Problem' and 'Optimization Problem', as well as their subsequent subclasses. If the user selects a filter value, e.g., "optimization", all data nodes directly connected by the property *rdf:type* with the concept 'Optimization Algorithm' or indirectly related through *rdf:type* with one of its subclasses, are retrieved. The successful retrieving of search results for these facets requires prior unique definitions of the concepts involved, e.g., the concept 'Optimization Algorithm' should be defined as 'Optimization Algorithm $\equiv$ Algorithm $\sqcap \exists$ solves.Optimization Problem'. The advantage of these facets is the ability to enhance IR with hierarchical dependencies. However, due to requirements for automatic inferences and deep graph querying, they can be time-consuming. To avoid deep queries by the extraction of broader terms, we restrict the query to the maximal depth d. To reduce the inference time, we deduce implicit knowledge about the type of instances, e.g., whether a specific algorithm is an optimization algorithm, during the data integration into the KG. Yet, this approach does not guarantee discovering all potential inferences related to new or modified instances. Hence, after n data insertions into the KG, we extract the graph and employ the Hermit [18] reasoner to derive additional assertions throughout the entire graph. *Property facets* rely on the range of specific relations. These facets are useful when the values are structured as a flat tree. In contract to semantic type facets, there is no need to define new concepts and corresponding definitions. An example is the "software tool" facet with values, e.g., "Qiskit", or "Pennylane". The results are retrieved by filtering out the instances that are connected by the relation *qco:depensOn* with the selected facet value.

Faceted search on the PlanQK platform starts with opening the advanced search interface for a specific core entity. This action triggers the generation of the facets and their possible values. Figure 3 depicts this process. To enable the construction of the SPARQL query for retrieving available facet values, we read the search configuration for the selected entity. This configuration specifies following attributes for each facet: (i) facet name, (ii) URI(s) of top node(s), and (iii) facet type (semantic type vs. property facet). As the structure of the facets depends on the underlying model, we query here the triple store directly. After the tree of facets is displayed, the user starts choosing desired filters. As shown in

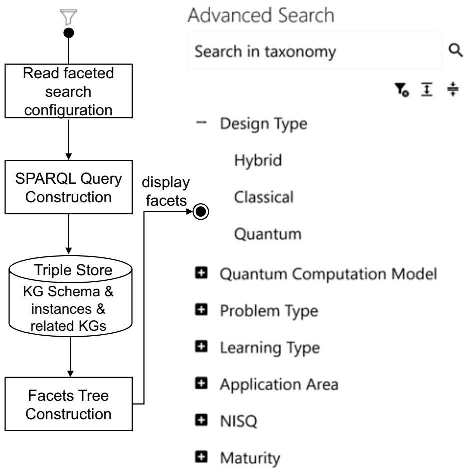


Figure 3. Generation of the facets tree.

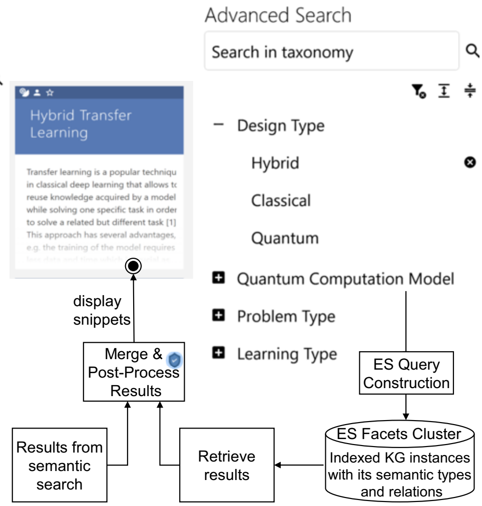


Figure 4. Retrieval of results in the faceted search.

Figure 4, we query the corresponding ES cluster to retrieve results, which holds only indexed search related information. In the post-processing step, the results from the faceted search (i) are merged with the result set from the semantic search (if it was executed first), and (ii) filtered to match user access rights for the content. Final results are displayed as snippets in the frontend.

4. Conclusion and Outlook

Curated KGs, which undergo a careful engineering and data integration, act as useful resources for various applications such as IR, decision support and recommendation systems. In this work, we presented the pipeline for an on-the-fly integration of the PlanQK platform data into a unified KG. The semantification process can be further improved by extracting relevant concepts and relations from textual attributes through machine learning enhanced techniques, e.g., named entity recognition. To ease IR over the platform, we implemented the native semantic search as well as the faceted semantic search. Further work is required to evaluate the search functionality and refine the search capabilities. We plan to expose a part of PlanQK KG as Linked Open Data and establish connections with related external sources such as DBPedia [19], Wikidata [20] and graphs in the research domain, e.g., Open Research Knowledge Graph (ORKG) [21]. As the knowledge in the field of quantum computing is currently changing rapidly, the possibility of the controlled graph curation by the community should be established.

Acknowledgments

This work was partially funded by the BMWK project *PlanQK (01MK20005F /01MK20005A)*.

References

[1] Peng C, Xia F, Naseriparsa M, OsborneF. Knowledge Graphs: Opportunities and Challenges. Artificial Intelligence Review; 2023 Apr; p. 1 - 32.

[2] Hogan A, Blomqvist E, Cochez M, d'Amato C, de Melo G, Gutiérrez C, Kirrane S, Gayo JE, Navigli R, Neumaier S, Ngomo AN, Polleres A, Rashid SM, Rula A, Schmelzeisen L, Sequeda J, Staab S, Zimmermann A. Knowledge Graphs. ACM Computing Surveys (CSUR). 2021 July; 54(4): 1 - 37.

[3] Harris S, Seaborne A, Prud'hommeaux E. SPARQL 1.1 Query Language, W3C Recommendation. W3C Recommendation. World Wide Web Consortium [Internet]. 2013 Mar [cited 2023 May 30th]. Available from: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[4] Martyniuk D, Falkenthal M, Karam N, Paschke A, Wild K. An analysis of ontological entities to represent knowledge on quantum computing algorithms and implementations. In: Paschke A, Rehm G, Al Qundus J, Neudecker C, Pintscher L, editors. Proceedings of the Conference on Digital Curation Technologies; 2021 Feb 8-12; Berlin, Germany.

[5] Publio GC, Esteves D, Lawrynowicz A, Panov P, Soldatova LN, Soru T, Vanschoren J, Zafar H. ML-Schema: Exposing the Semantics of Machine Learning with Schemas and Ontologies. 2018; ArXiv, abs/1807.05351.

[6] FAIRsharing.org. SRAO, Subject Resource Application Ontology [Internet]. 2018 [cited 2023 May 30th]. Available from: https://fairsharing.org/FAIRsharing.b1xD9f.

[7] Software package data exchange SPDX License List [Internet]. 2023 [cited 2023 May 30th]. Available from: https://spdx.org/licenses/ (Accessed: 29 May 2023).

[8] Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018 Aug; 2: 79.

[9] Endo S, Cai Z, Benjamin SC, Yuan X. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. Journal of the Physical Society of Japan. 2021 Feb; 90 (3).

[10] Weder B, Barzen J, Beisel M., Leymann F. Provenance-Preserving Analysis and Rewrite of Quantum Workflows for Hybrid Quantum Algorithms. SN Computer Science, SCI. 2023 Feb; 4 (233).

[11] Knoblock CA, Szekely P, Ambite JL, Goel A, Gupta S, Lerman K, Muslea M, Taheriyan M, Mallick P. Semi-automatically mapping structured sources into the semantic web. In: Simperl E, Cimiano P, Polleres A, Corcho O, Presutti V, editors. Proceedings of the 9th international conference on The Semantic Web: research and applications. ESWC 2012; Heidelberg; 7295: 375–390.

[12] Karam N, Müller-Birn C, Gleisberg M, Fichtmüller D, Tolksdorf R, Güntsch A. A terminology service supporting semantic annotation, integration, discovery and analysis of interdisciplinary research data. Datenbank-Spektrum. 2016 Mar; 16: 195 – 205.

[13] Zhang X, Hou X, Chen X, Zhuang T. Ontology-based semantic retrieval for engineering domain knowledge. Neurocomputing. 2013 Sep; 116: 382-391.

[14] Sharma D, Pamula R, Chauhan DS. Semantic approaches for query expansion. Evolutionary Intelligence. 2021 Mar; 14: 1101–1116.

[15] Guha R, McCool R, Miller E. Semantic search. In: Hencsey G, White B, editors. Proceedings of the 12th international conference on World Wide Web. 2003 May; p. 700–709.

[16] Grau BC, Kharlamov E, Marciuska S, Zheleznyakov D, Arenas M: SemFacet. Faceted Search over Ontology Enhanced Knowledge Graphs, International Workshop on the Semantic Web. In: Kawamura T, Paulheim H, editors. Proceedings of the ISWC 2016 Posters and Demonstrations Track. 2016 Oct; Kobe, Japan.

[17] Heidari G, Ramadan A, Stocker M, Auer S. Demonstration of Faceted Search on Scholarly Knowledge Graphs. In: Leskovec J, Grobelnik M, Najork M, Tang J, Zia L, editors. WWWW '21: Companion Proceedings of the Web Conference. 2021 Apr; p. 685–686 .

[18] Glimm B, Horrocks I, Motik B, Stoilos G, Wang Z. HermiT: An OWL 2 Reasoner. Autom Reasoning. 2014 May; 53: 245–269.

[19] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. Dbpedia: a nucleus for a web of open data. In: The semantic web. 2007; p. 722–735.

[20] Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Communications of the ACM. 2014 Sep; 57(10): 78–85.

[21] Jaradeh MY, Oelen A, Prinz M, Stocker M, Auer S. Open Research Knowledge Graph: A System Walkthrough. In: Doucet A, Isaac A, Golub K, Aalberg T, Jatowt A, editors. Digital Libraries for Open Knowledge. Lecture Notes in Computer Science, Springer; 2019; 11799.

Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

Subject Index

This page intentionally left blank

Author Index