Improving maintenance of community-based knowledge graphs

Nicolas Ferranti[1][0000−0002−5574−1987]

Vienna University of Economics and Business, Welthandelspl. 1, 1020 Vienna, Austria
nicolas.ferranti@wu.ac.at

Abstract. Data quality is crucial for the effective utilization of knowledge graphs, ensuring it is challenging due to the need for continuous monitoring and maintenance. This research proposal focuses on data quality in open knowledge graphs, with an emphasis on Wikidata. Wikidata, one of the largest collaborative knowledge graphs, has its own approach for data consistency, deviating from regular OWL ontologies or SHACL, the W3C recommendation. The proposal aims to comprehend and formalize Wikidata's approaches for assessing and resolving data inconsistencies. By formalizing constraints, refinement operations, and repair strategies, this research aims to improve the quality of Wikidata and other knowledge graphs also developed based on Wikibase. As one of the contributions, our research proposes a semi-automatic refinement pipeline to empower the Wikidata user community by recommending repairs of constraint violations, combining distance-based refinement approaches and ranking heuristics. Establishing a comprehensive framework and engaging users in knowledge graph maintenance enhances the reliability and usability of open knowledge graphs.

Keywords: Data Quality · Knowledge Graph · Wikidata · Refinement.

1 Introduction

A Knowledge Graph (KG) uses a graph-based model to represent real-world entities, their attributes, and relationships [11]. Entities can be anything that can be uniquely identified and described, such as people, places, things, or concepts. Statements, representing relationships between entities are represented as edges in the graph, while the attributes of entities are again graph nodes. As such, KGs can be used to represent a wide range of information, including encyclopedic knowledge, scientific data, or enterprise information [8].

There are different methods that can be used to create a KG. For instance, they can be extracted from semi-structured Web data, like DBpedia KG [9], or edited collaboratively by a community, like the Wikidata KG [16]. Regardless of the approach used in construction, KGs are not perfect. Despite the efforts of some communities or organizations to increase the coverage of their respective KGs, it is complex to represent all the knowledge available about a domain. Therefore, organizations responsible for KGs usually look for a balance between correctness and information coverage.

Since its creation by the Wikimedia Foundation in 2012, Wikidata (WD) has become one of the largest KGs, publicly available on the Web, with more than 100M items[1] and 14B triples[2]. One of the main factors responsible for this growth is WD's user community, with more than 24k active users (humans and bots). The large user community is primarily motivated by Wikipedia, as the vast majority of Wikipedia pages incorporate content from WD [3]. The WD KG is available in standard RDF format and can be queried via a public SPARQL endpoint, but it does not adhere to established W3C standards, such as OWL/RDFS or SHACL for express constraints on its schema: unlike other knowledge graphs, WD focuses on the development of the data layer (A-box), and the terminology layer (T-Box) evolves alongside it without a predefined formal ontology [14].

Problem statement. In our proposal, we address the problem of data quality in KGs with an emphasis on WD. WD uses constraints to ensure consistent vocabulary usage, whereby WD projects do not currently utilize SHACL for validating RDF graphs against constraints as recommended by W3C, nor OWL ontologies. Studies to understand the semantics behind the WD constraint projects and allow violations to be easily tested and consumed by third parties are lacking. Our hypothesis is that such studies could leverage the development of approaches to assist users in refining inconsistent data, as well as promote constraint checking in WD through different approaches, such as by mapping constraints to SHACL and SPARQL. Currently, data edits are done manually and without a support tool for the community, which increases the effort to correct inconsistent data.

Contributions. In this proposal, our goal is to initially study the semantics of constraints used by WD, among the main contributions is the formalization of constraints in both SHACL and SPARQL languages, consequently creating one of the biggest benchmarks for SHACL validators and allowing other agents to retrieve inconsistent data in real-time through WD's SPARQL endpoint. The formalization of constraints is considered as the first step to make the following contributions possible:

- Efficient retrieval of inconsistent data
- Mapping of historical repairs
- Creation of models for proposing repair suggestions based on inconsistencies and repairs
- Use of knowledge graph embeddings as a distance-based refinement model
- Introduction of algorithms to improve the ranking of refinement suggestions according to different criteria, such as terminology data prevailing over instances or minimizing changes

[1] https://www.wikidata.org/wiki/Wikidata:Statistics, as of January 2023

[2] https://short.wu.ac.at/7t66, last accessed 13 February 2013

- Development of a pipeline to combine multiple refinement models with ranking strategies, for KG editing and maintenance

An overview of the main contributions grouped by the proposed workflow is available in Figure 1

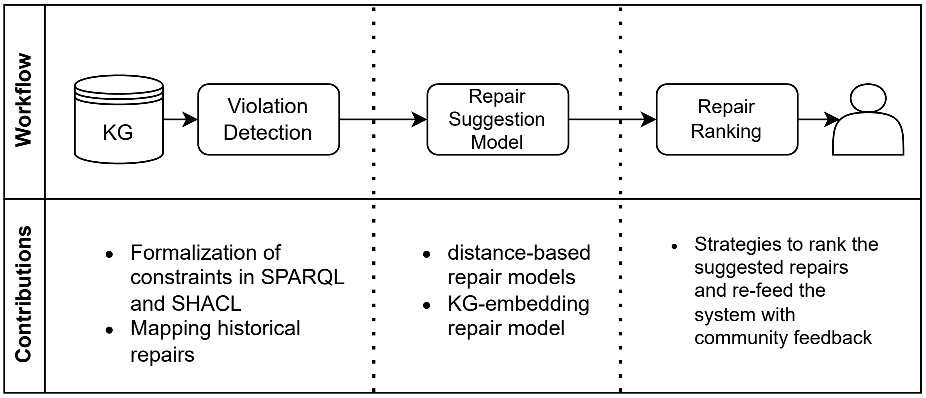
Workflow
KG
Violation Detection
Repair Suggestion Model
Repair Ranking
Contributions
• Formalization of constraints in SPARQL and SHACL
• Mapping historical repairs
• distance-based repair models
• KG-embedding repair model
• Strategies to rank the suggested repairs and re-feed the system with community feedback

Fig. 1. Main contributions grouped by the workflow step they refer to.

Paper structure. The remainder of this research proposal is structured as follows. Section 2 presents related work focusing on data quality and constraints formalization. Section 3 presents the main research questions and hypotheses organized in steps. Section 4 discuss some of the preliminary results concerning the formalization of WD constraints. In section 5, the methodology for our next steps based on the current findings is introduced. Finally, section 6 points to the main future directions.

2 Related Literature

Ensuring the quality of the data in a KG is fundamental for useful consumption. Paulheim [11] provided a comprehensive survey of KG refinement approaches, further classified according to their goal: KG completion, or repair detected errors. These approaches apply different methods, ranging from techniques using machine learning to NLP-related techniques. The results showed that the vast majority of approaches focused on DBpedia, indicating a gap when it comes to WD and other Wikibase-based KGs.

Furthermore, Shenoy et al. [15] presented a quality analysis of WD focusing on correctness, checking for weak statements under three main indicators: constraint violation, community agreement, and deprecation. The premise is that a statement receives a low-quality score when it violates some constraint, highlighting the importance of constraints for KG refinement. Shenoy and colleagues use a subset of constraints to retrieve violations through a toolkit and also mention the challenges of testing WD constraints using SHACL. However, the focus of the authors is on combining violation indicators with other indicators toward a quality metric, rather than formalizing the semantics of the constraints and promoting practical means for testing and repairs.

Martin and Patel-Schneider [10] discuss the representation of WD property constraints through multi-attributed relational structures (MARS), as a logical framework for WD. Constraints are represented in MARS using extended multi-attributed predicate logic (eMAPL), providing a logical characterization for constraints. Despite covering 26 different constraint types, the authors have not performed practical experiments to evaluate the accuracy of the proposed formalization, nor its efficiency.

Ahmetaj et al. [1] propose an approach to provide refinements to SHACL violations. The approach involves encoding the problem as an Answer Set Programming (ASP) program. By transforming the graph and a set of SHACL shapes into the ASP program P, the answer sets or stable models of P represent possible repairs. The use of efficient ASP solvers, such as [7], offers a promising means to generate practical data repairs. One of the major benefits of formalizing WD constraints into SHACL is to enable the use of the various solutions already implemented for SHACL constraints in the context of WD

In conclusion, the existing literature on KG refinement approaches has primarily focused on KGs like DBpedia, leaving a gap in understanding and refining KGs such as WD and other Wikibase-based KGs. While previous studies have highlighted the importance of constraints in KG quality analysis, there is a need to formalize the semantics of these constraints and establish practical means for testing. By formalizing constraints, violations and repairs can be collected and serve as valuable input for refinement models.

3 Research Questions and Hypotheses

Building on the existing challenge of refining community-based KGs, we see the need for semi-automatic refinement approaches able to provide the community with repair suggestions based on both formal definitions of constraints and statistical analysis. Inconsistencies are the primary input for performing corrective repairs. As pointed out by [15], WD inconsistency reports are calculated within an ad-hoc extension of Wikibase. On top of that, the approach behind the generation of the reports is not public, inconsistency reports are published on an HTML page with a maximum limit of inconsistencies displayed by each constraint type. Therefore, it is crucial to formalize the constraints and create an open and efficient method to retrieve inconsistencies.

In the recent past, embedding approaches have been used to address KG completion and link prediction. Bordes et al.[2] introduced embeddings for KGs. We would like to test whether these approaches and AI models trained based on identified inconsistencies and repairs can provide the community with relevant repair suggestions.

To this end, we summarise the main hypothesis of our research proposal as follows:

Effective maintenance of community-based KGs can be achieved by: (i) formally defined constraints in languages that optimize the process of collecting inconsistent data; (ii) a set of approaches to propose refinement suggestions using inconsistent data and previous repairs as input; (iii) a set of heuristics to rank candidate repairs according to different preferences.

Our hypothesis leads to the following research questions:

1. Can the semantics of Wikidata property constraints be represented with SHACL-core and SPARQL?
2. How can we make use of inconsistencies and historical repairs to propose refinements to knowledge graphs?
3. How can distance-based metrics be used to propose/predict repairs to the knowledge graph?
4. What are the most relevant strategies to rank different repair suggestions?

4 Preliminary results

In an effort to understand the semantics, formalize, and operationalize WD property constraints, we first investigated, based on available descriptions, whether and how the 30 WD property constraints could be mapped to SHACL's core language and SPARQL [5]. This study made it possible to clarify to which extent SHACL can represent community-defined constraints of a widely used real-world KG. One of our results is a collection of practical SHACL constraints that can be used in a large and growing real-world dataset; indeed the non-availability of practical SHACL performance benchmarks has already been emphasized by [6].

Other results we presented include clarifications of heretofore uncertain issues, such as the representability of permitted entities and exceptions in WD property constraints within SHACL [15]. We also could argue the inexpressibility of certain WD constraints, due to the impossibility to compare values obtained through different paths matching the same regular path expression within SHACL-core. These issues could be addressed when using SPARQL to formalize and validate constraints, where all 30 constraints could in principle be formalized.

Subsequently, we compared the inconsistencies found by the new constraint set against the primary reference, the inconsistency reports system[3]. In a recently submitted journal paper [4], we identified that 5 constraint types represent the vast majority of reported inconsistencies, therefore an experiment was designed to compare the top five properties of each constraint type against the results obtained by our SPARQL formalization. The results summarized in Table 1

[3] https://www.wikidata.org/wiki/Wikidata:Database_reports/Constraint_violations

and detailed in [4] show that there is a difference in inconsistencies identified when an approach considers only the truthy statements from when it considers the whole statements set. Furthermore, we observed that there is still room for discussions on how these constraints should be checked due to the absence of formal, unambiguous specifications in some cases and also due to the existence of qualifiers not considered by WD's report tool when validating the data.

Our first research question can be partially answered, as we found that SPARQL can serve as a viable solution to retrieve inconsistent data in real-time, as well as stimulate discussion to consolidate the semantics of each constraint. Similar tests using SHACL-core are also under execution.

	WD inconsistency reports		**# found by our**	
Constraint Type	**# reported**	**# available (RA)**	**SPARQL (SPA)**	**(RA ∩ SPA)**
One-of	2390363	25005	1413369	20122
IRS	1241897	79812	1246789	79812
Single-value	236851	25152	148792	15475
Required Qualifier	809174	26023	807300	26023
VRS	1955726	33418	1949745	33379

Table 1. Summary of the inconsistencies compared. RA ∩ SPA represents the intersection between inconsistencies provided by the reports page and those found by the SPARQL query. Abbreviations used: Constraint Type (IRS = Item-requires-statement, VRS= Value-requires-statement). Details in [4].

5 Methodology for next steps

To address our research questions, we employ two research methodologies: (i) a systematic literature review on KG refinement approaches; and (ii) the design science research methodology (DSRM) put forth by [13]. While the first contributes to understanding the state of the art in refinement approaches, the second can be used to consume the set of inconsistencies and repairs and develop innovative refinement technologies. Bellow, DSRM activities are described, as well as how our research proposal fits in them.

Problem and motivation. The research is driven by practical problems and aims to develop solutions that address specific challenges faced in practice. It focuses on creating artifacts or designs that have value and utility in solving real-world problems. In the scope of this research, the problem to be solved is to facilitate the process of corrective editing in collaborative KGs by developing a semi-automatic tool to promote repair suggestions.

Design and Creation. DSRM involves the design and creation of new artifacts, such as models, methods, frameworks, or software prototypes. These artifacts are intended to improve or enable certain aspects of a given problem domain. Through a systematic review of the literature, we can identify the main refinement methods and propose the use of inconsistencies and historical repairs in the creation of a distance-based semi-automatic refinement tool. For instance, making use of KG embedding methods to assess the distance of the violating value to the vector that describes the expected values. This method, in combination with the consumption of historical repairs, can help to predict what would be an optimal repair and present options for the community.

Evaluation. The designed artifacts are rigorously evaluated to assess their effectiveness, efficiency, and utility in solving the identified problems. The evaluation process often includes usability testing, performance measurement, and gathering feedback from relevant stakeholders. It is intended that the artifacts developed in this research can be tested both to predict corrections based on historical data and on the level of interaction with KG user editors. Therefore, two main experiments are expected, one over a benchmark of historical repairs and, once the tool is fully operational, a qualitative study with the WD user community.

6 Reflection and Future Work

In this research proposal, we explored the problem of refining community-based KGs through the formalization of constraints and the usage of inconsistencies as input for the creation of semi-automatic distance-based refinement approaches.

We note that the main approaches do not focus on community-based KGs [12], where we plan to contribute with a systematic review analyzing more recent studies. Due to the fact that Wikidata, the most popular community-based KG, does not use conventional methods such as OWL ontologies to represent data terminology [14], our first efforts focused on formalizing constraints created by the community itself [4, 5]. Our next steps consist of analyzing sets of violations and identifying repair patterns in KG to build semi-automatic repair approaches. In the future, we hope to build a solution capable of suggesting relevant fixes to the community according to different objective functions.

Acknowledgements This research is conducted under the supervision of Prof. Dr. Axel Polleres.

References

1. Ahmetaj, S., David, R., Polleres, A., Šimkus, M.: Repairing shacl constraint violations using answer set programming. In: The Semantic Web–ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. pp. 375–391. Springer (2022)

2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 2787–2795 (2013)
3. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: International semantic web conference. pp. 50–65. Springer (2014)
4. Ferranti, N., De Souza, J.F., Polleres, A.: Formalizing and validating wikidata's property constraints using shacl+ sparql https://www.semantic-web-journal.net/system/files/swj3378.pdf, note: Under review
5. Ferranti, N., Polleres, A., de Souza, J.F., Ahmetaj, S.: Formalizing property constraints in wikidata. In: Proceedings of the Wikidata Workshop co-located with 21st International Semantic Web Conference (ISWC, 2022), Hangzhou, China, October 23-27, 2022 (2022)
6. Figuera, M., Rohde, P.D., Vidal, M.: Trav-shacl: Efficiently validating networks of SHACL constraints. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 3337–3348. ACM / IW3C2 (2021). https://doi.org/10.1145/3442381.3449877, https://doi.org/10.1145/3442381.3449877
7. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Multi-shot asp solving with clingo. Theory and Practice of Logic Programming **19**(1), 27–82 (2019)
8. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.d., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. Synthesis Lectures on Data, Semantics, and Knowledge **12**(2), 1–257 (2021)
9. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
10. Martin, D., Patel-Schneider, P.F.: Wikidata constraints on mars. In: Wikidata@ ISWC (2020)
11. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
12. Paulheim, H., Gangemi, A.: Serving dbpedia with dolce–more than just adding a cherry on top. In: International semantic web conference. pp. 180–196. Springer (2015)
13. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of management information systems **24**(3), 45–77 (2007)
14. Piscopo, A., Simperl, E.: Who models the world? collaborative ontology creation and user roles in wikidata. Proceedings of the ACM on Human-Computer Interaction **2**(CSCW), 1–18 (2018)
15. Shenoy, K., Ilievski, F., Garijo, D., Schwabe, D., Szekely, P.: A study of the quality of wikidata. Journal of Web Semantics **72**, 100679 (2022)
16. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)