Robust Knowledge Graph Cleaning

Maximilian K. Egger[1]

Supervised by: Davide Mottin[1] and Panagiotis Karras[1,2]
[1]Aarhus University, Nordre Ringgade 1, 8000 Aarhus C, Denmark
[2]Copenhagen University, Nørregade 10, 1172 Copenhagen, Denmark

Abstract

Data quality is needed to properly and reliably use the information represented in the dataset. The increasing volume of data renders data preparation and cleaning increasingly difficult. Additionally, more diverse types of data structures for databases, like graphs, get used and need to be handled differently. This leads to the necessity of robust methods to increase data integrity, scalable approaches for finding and fixing errors, and local-oriented algorithms that can be used to pinpoint attention where needed. In my PhD project, I focus mainly on knowledge graph structures and define and establish different tools that can be used to clean the knowledge graphs.

Keywords

Knowledge Graphs, Data Mining, Data Quality

1. Introduction

In today's data-driven world, information and knowledge are mined, processed, and used in almost any digital setting. With the rise of machine learning and artificial intelligence in many daily applications, most individuals are affected by their reliability and accuracy in their respective tasks. Therefore, it is essential that the information and data are factually correct, if available, or as accurate as possible. One framework for interacting and working with such data is *Knowledge Graphs* (KGs) [1]. A general KG is a network of heterogeneous information of *entities* that are connected with *relationships* [2]. Entities (nodes) are objects that are either representations from real life, like people or places, or abstract concepts. Relations describe the relationships such objects have with each other. Additionally, there are types and categories that can be applied to entities and relations. A complete directed connection between two entities with a specific relation is called a *fact* or *triple.* If nodes and relations also have properties themself, it is considered a property graph [3].

KGs are applied in different disciplines of research, such as medicine [4], social sciences [5], and drug discovery [6]. In daily life, they are also used when searching the web via Google [7], often unbeknownst to the regular user. If there is a box to the right of your search result page, it is a response generated by the KG of Google. These panels provide factual information on the respective search terms.

KGs can be utilized by experts in the domain to query for specific information that they require to further their own research. A current highly relevant use case is the drug discovery process to save time and money for developing new medicines. In these projects, it is possible to model the benefits and side effects of various drug elements before synthesizing them in the lab for clinical studies [8].

Currently, with the rise of interest in generative AI like Chat GPT, Gemini, and Copilot, the generation of factual incorrect but plausible-sounding information has become a lot easier; this gives a need for factual correct answers. KGs are one possible solution to aid LLMs with the framework of retrieval augmented generation (RAG)[9]. This process allows the respective LLM to query for a factual answer in the KG to support the generated answer to the user. If done correctly, the generated answer will then contain the answer from the KG, which is correct, given that the underlying KG has no error.

In all of the mentioned use cases, errors in the results or the knowledge extracted in the KG can result in higher costs in time and resources.

In my PhD, I explore the notion of robustness as the ability of a knowledge base to work as intended even in the presence of incomplete, erroneous, redundant, and inconsistent data and accommodate such data in a way that reduces incompleteness and eliminates errors, redundancies, and inconsistencies. Towards this, I aim to solve the following research questions:
(RQ1) Is there a measure that provides a prior indication of the reliability of a KGE on a specific subgraph?
(RQ2) Are there normal forms for graphs that can increase the data integrity?
(RQ3) Are there logical rules that can be found and utilized on topic-based subgraphs?

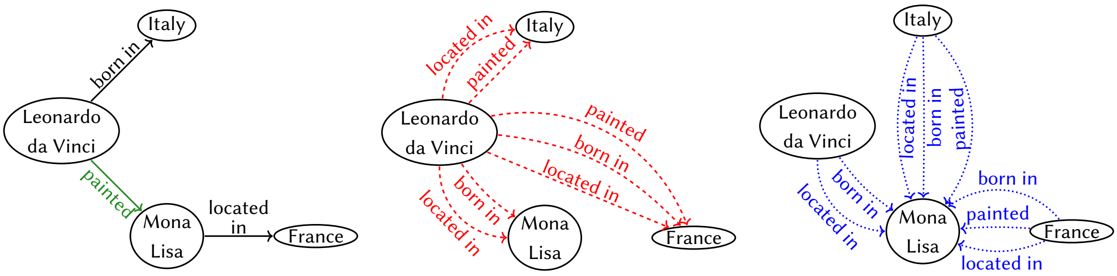
This paper presents the work done in the first two years of my PhD as well as some ideas for my future work. The structure is as follows. Section 2 covers the related work regarding our approaches to support knowledge graph cleaning; Section 3 covers the main contributions; Section 4 presents future work and challenges; Section 5 concludes the paper.

2. Related Work

Knowledge graph cleaning is the focus of several research areas. Here, we restrict our focus to the immediate areas regarding my projects from the first two years as well as my planned future work.
Knowledge graph embeddings (KGEs) are used commonly for various tasks, like detecting missing triples, correcting errors, or question answering [10, 11]. There are several different KGE types and fitting examples like, *Translational embeddings* (TansE [12]), *Semantic embeddings* (DistMult [13]), *Complex embeddings* (ComplEx [14]) and *Neural-network embeddings* (ConvE [15]).
Evaluation of embeddings is mainly done with ranking-based measures, in particular with HITS@k and mean reciprocal rank (MRR) for head, tail, and relation prediction [10, 16, 17]. These measures indicate performance globally, but so far, no measure provides local analysis capabilities.

Published in the Proceedings of the Workshops of the EDBT/ICDT 2025 Joint Conference (March 25-28, 2025), Barcelona, Spain

maximilian.egger@cs.au.dk (M. K. Egger)
0009-0001-4473-4984 (M. K. Egger)



KG and considered edge

Negative triples $\mathcal{N}^-(h)$ to compute $rank_H$

Negative triples $\mathcal{N}^-(t)$ to compute $rank_T$

Figure 1: Constituents of *ReliK* on an example KG.

Functional dependencies (FDs) define directed relationships between attributes in the data. Therefore they are used as blocks to construct keys and normal forms [18]. Similarly, FDs tailored to graph models are pairs of a graph pattern and an implication [19, 20, 21].
Data normalization uses normal forms (NF) to reduce data redundancies in the chosen representation, like first, second, third, and BC normal form [18] for relational databases. BCNF has been extended towards XML documents and their underlying tree structure [22, 23, 24] as a first step toward general graphs. A recent attempt at *graph normalization* has been made that uses uniqueness constraints [25] and graph-tailored functional dependencies [26] that target node properties. This leaves a gap of NFs that handle all parts of a graph and are not zoned in on the node level.
Rule mining in KGs finds logic rules that can be used to find and fill in missing information throughout the data [27, 28]. These also provide human-readable statements that can be used for the reasoning process to fill in the graph. These rules are evaluated and constructed from a global perspective, which could lead to topic-specific rules being overlooked that are still relevant and valid in their respective contexts.

3. Contributions

Here, we first discuss our completed work on reliability in knowledge graph embeddings and then introduce our current endeavors on normal forms for graphs. In the initial project *ReliK*, we defined and evaluated a new metric for the local reliability of KGEs. In the second project, which is in the process of being submitted, we looked into property graphs and established normal forms for them to reduce data redundancy.

3.1. ReliK

KGEs are heavily used for a variety of *downstream tasks* that rely on the underlying KG being complete and the KGE being well trained. Their evaluation so far has only been done on a global scale with their respective tasks in mind. Therefore, an open problem is a more general metric that can indicate behavior independent of the application while also being unprejudiced towards the chosen embedding model or the underlying data.

These issues can be addressed by our measure *ReliK* [29], which is a straightforward yet principled approach that assesses the *reliability* of a KGE's performance on a specific downstream task within a particular section of the KG, all without executing the task or (re)training the KGE. *ReliK* only relies on the existing embedding scores as a black box. These scores are only used to create a ranking that is fed into our measure.

Specifically, two rankings are used to get the value for a triple. Figure 1 shows what is considered to be part of the respective ranking. Namely, the two negative neighborhoods that are used to measure the triple against. The negative neighborhood aimed at the head (h) part of the triple consists of all triples with the form $(h, ?, ?)$ that are not part of the original KG. For the tail (t), this is done in a similar manner. Then, the embedding score for the neighborhoods and the correct triple x_{hrt} is evaluated, and the ranking is established. This gets put into the following formula to constitute the ReliK score.

$$ReliK(x_{hrt}) = \frac{1}{2}\left(\frac{1}{rank_H(x_{hrt})} + \frac{1}{rank_T(x_{hrt})}\right).$$

This can also be extended to a subgraph level by taking the mean of the respective ReliK scores for all triples in the subgraph.

Consequently, *ReliK* is agnostic to (1) the specific characteristics of a given KGE, (2) the particular KG in question, and (3) does not require any KGE retraining. Furthermore, (4) *ReliK* is task-agnostic: its design principles are so broad that it is naturally suited for a wide range of downstream tasks for more details. Finally, (5) *ReliK* possesses the locality property, allowing its computation and semantics to be tailored to specific parts of the KG. Overall, our *ReliK* measure fully meets all the aforementioned criteria. It is also important to note that *ReliK* can be utilized to evaluate the effectiveness of a KGE for a downstream task, even when we only have access to the embeddings for privacy or other reasons, rather than the original KG.

ReliK is simple, intuitive, and easy to implement. Despite that, its exact computation requires processing all the possible combinations of entities and relationships for every single fact of interest. Therefore, we also introduced two approximations to calculate a good estimate of the exact *ReliK* for large KGs. One of them is a good approximation in expectation, while the other is a strict lower bound of the original *ReliK* if this is needed for theoretical guarantees.

To showcase that both of these approximations work as expected, we present both runtime and MSE for a small dataset in which the calculation of the accurate *ReliK* is feasible.

To verify our metric and its approximations, we have conducted an extensive study in which we evaluate against measures like MRR for tail, relation, and triple classification, as well as more complicated tasks with query answering and rule mining.

The results of the experiments support that *ReliK* correlates with the accuracy of the prediction and classification tasks, which provides deeper insight into the reliability of

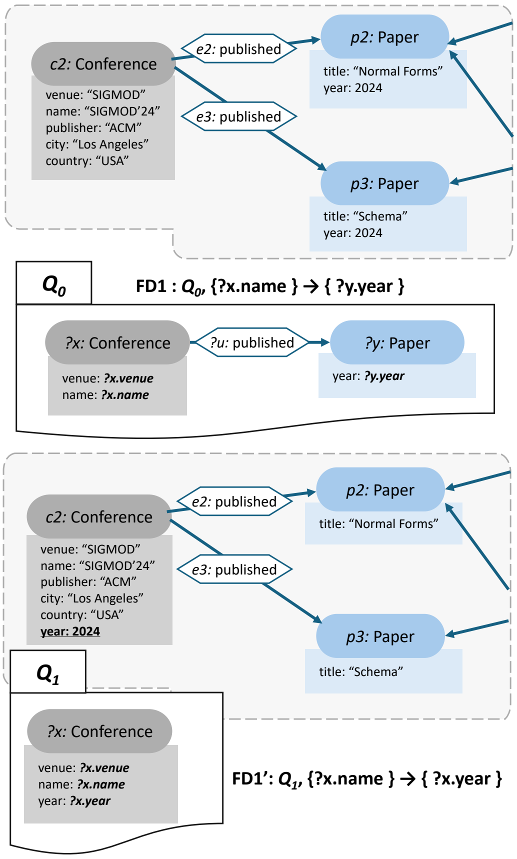


Figure 2: Fragments of graph satisfying 1GNF and 2GNF, with associated patterns and GFDs.

KGEs. Especially *ReliK* is able to differentiate between the correct and incorrect rule instances for complex logical rules as well as guiding the rule mining in subgraphs towards high-confidence rules.

3.2. Graph normal forms

Graph databases like property graphs do have sources of redundancies like any database can have. This issue can be resolved in relational databases or XML files via the concept of normal forms (NF) and a process to transform the original data into that format.

For graph databases, this does not exist in the same format yet. Still, the foundations like graph functional dependencies [19, 20, 21] and a first attempt that focuses on nodes have been made [26].

In our project, we establish a set of five graph normal forms (GNF) that use and build on top of these ideas by considering the complete graph in its structure. The process of utilizing edges and their properties cannot be done trivially from the NFs of the relational setting.

To define the GNFs, we use graph functional dependencies (GFDs) that consist of a pattern and a one-way dependency between two sets of attributes. An example of such can be seen in Figure 2.

1GNF disallows implicit links between nodes that could be represented by regular edges and nested attributes that hide data complexity. Increasing from there, the 2GNF forbids the replication of attribute values that are dependent on the key of a different node; 3GNF disallows partial dependencies from a key towards attributes; 4GNF only allows attributes directly dependent on a superkey; EGNF removes all value duplication by enforcing that every property is a key.

We provide algorithms that transform any property graph into its respective GNF versions without losing any information that may be stored in the database. One example of how a graph not in 2GNF can be transformed into one can be seen in Figure 2. In this small example, the attribute *year* from paper nodes can be connected to the *name* of conference nodes with the GFD FD1, which states that the publish *year* of a paper is determined by the *name* of the conference it was published in. This is a violation of the 2GNF, and in order to remove the violation, the attribute *year* is moved into the conference node. When done for all conference and paper nodes, this reduces possible redundancy throughout the graph and preserves information. It should be noted that to query for the same piece of information, a different query is needed between the original snippet and the 2GNF version of it.

To show that our established GNFs do reduce data redundancy, we perform experiments in which we count the total number of attribute values in a selection of datasets, perform the transformations into the GNFs, and then count in the transformed datasets. The number of attribute values is getting lower as more GNFs are applied, thus showing that GNFs are able to increase the data integrity by reducing redundancy in graphs. The process of transforming into the different GNFs incurs the addition of new edges and nodes to the dataset to facilitate the changes needed for the consolidation of information.

4. Future Work

The next avenue to look into for graph cleaning for my PhD project is rule mining. Specifically, the problem of contextual rule mining is about the possibility of rules having context and a local neighborhood of validity. So far rule mining has been nearly exclusively done on a global scale [27, 30, 28, 31, 32], this could lead to topic-specific rules to be overlooked in the process. Additionally focusing on a specific area of a KG to mine rules opens up the chance of generating higher complexity rules and structures.

Just selecting random subgraphs probably does not suffice. Such subgraphs should capture different contexts like domain, temporal, or geographical areas. Therefore a new method, similar to community detection, will be needed to avoid human-heavy annotation of datasets. Especially since topic areas will not necessarily be strongly connected to communities in KGs.

Another challenge is guaranteeing significance and statistical support for the rules based on a smaller search space. Here, the absolute support of a rule in a subgraph will be at most equal to the global setting, which leads to a trade off between subgraph size and statistical significance.

To motivate the validity of contextual rule mining, we report some preliminary experiments on a subset of the Freebase dataset that has six annotated domains in the graph[1]. In Table 1 we see some results of applying the rule mining method AMIE [27] on both the complete set as well as exclusively the respective domains. In three of these topic area subgraphs we were able to find rules that are not found and presented in the entire set of these six domains. This observation supports the claim that these kinds of rules

[1] https://people.cs.aau.dk/~matteo/notes/freebase-data-dump.html

Domain	Triples	Rules	Dom. specific
complete	4302875	1636	-
organization	1767483	106	0
government	613575	405	17
military	260973	68	4
business	1408406	758	11
geography	139900	5	0
finance	112538	66	0

Table 1
Domain details in size and number of mined rules

exist in KGs. Further investigation of the relevance and significance of rules found in this approach is needed as to how to apply this concept on datasets that do not have domains pre-labeled.

5. Conclusion

In my PhD, I study the notion of robustness in knowledge bases. Towards this I investigate the reliability of knowledge graph embeddings, eliminating redundancies in graphs, and contextual rule mining.

Specifically, ReliK (1) gives the possibility of verifying which areas of knowledge graph embedding can and should be used in the cleaning and knowledge completion process. Enforced by an intuitive metric that can be applied independent of model choice. Graph normal forms (2) provide the needed reduction in data redundancy that increases data integrity, as well as giving a standardized way to normalize the data. Finally, I present locally aware rules (3) as a future project that can be used to get topic-specific rules, which can be further used to establish correctness in applicable subgraphs in a nuanced approach.

References

[1] A. Tchechmedjiev et al., Claimskg: A knowledge graph of fact-checked claims, in: In The Semantic Web–ISWC, Springer, 2019, pp. 309–324.

[2] C. Shi et al, A survey of heterogeneous information network analysis, TKDE 29 (2016) 17–37.

[3] R. Angles et al., Pg-schema: Schemas for property graphs, Proceedings of the ACM on Management of Data (2023) 1–25.

[4] L. Li et al., Real-world data medical knowledge graph: construction and applications, Artificial intelligence in medicine 103 (2020) 101817.

[5] M. Conti et al., A model to represent human social relationships in social network graphs, SocInfo, pages 174–187 (2012).

[6] F. MacLean, Knowledge graphs and their applications in drug discovery, Expert opinion on drug discovery 16 (2021) 1057–1069.

[7] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic web 8 (2017) 489–508.

[8] X. Zeng et al., Toward better drug discovery with knowledge graph, Current opinion in structural biology 72 (2022) 114–126.

[9] P. Lewis et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[10] Q. Wang et al., Knowledge graph embedding: A survey of approaches and applications, TKDE 29 (2017) 2724–2743.

[11] S. Ji et al., A survey on knowledge graphs: Representation, acquisition, and applications, Trans. Neural Netw. Learn. Syst. 33 (2021) 494–514.

[12] A. Bordes et al., Translating embeddings for modeling multi-relational data, NeurIPS 26 (2013).

[13] B. Yang et al., Embedding entities and relations for learning and inference in knowledge bases, in: ICLR, 2015.

[14] T. Trouillon et al., Complex embeddings for simple link prediction, in: ICML, PMLR, 2016, pp. 2071–2080.

[15] T. Dettmers et al., Convolutional 2d knowledge graph embeddings, in: AAAI, volume 32, 2018.

[16] T. Safavi et al., Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction, in: EMNLP, 2020.

[17] F. Bianchi et al., Knowledge graph embeddings and explainable ai, in: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020, pp. 49–72.

[18] E. F. Codd, Further normalization of the data base relational model, Data base systems 6 (1972) 33–64.

[19] W. Fan et al., Functional Dependencies for Graphs, in: SIGMOD, 2016, pp. 1843–1857.

[20] W. Fan et al., Capturing associations in graphs, VLDB (2020) 1863–1876.

[21] W. Fan et al., Discovering association rules from big graphs, VLDB (2022) 1479–1492.

[22] M. Arenas et al., A normal form for xml documents, TODS (2004) 195–232.

[23] M. Arenas et al., An information-theoretic approach to normal forms for relational and xml data, JACM (2005) 246–283.

[24] M. Arenas, Normalization theory for XML, SIGMOD (2006) 57–64.

[25] P. Skavantzos et al., Uniqueness constraints on property graphs, in: International Conference on Advanced Information Systems Engineering, Springer, 2021, pp. 280–295.

[26] P. Skavantzos et al., Normalizing Property Graphs, Proceedings of the VLDB Endowment 16 (2023) 3031–3043.

[27] L. A. Galárraga et al., AMIE: association rule mining under incomplete evidence in ontological knowledge bases, in: TheWebConf, 2013, pp. 413–422.

[28] L. Wu et al., Rule learning over knowledge graphs with genetic logic programming, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 3373–3385.

[29] M. K. Egger et al., Relik: A reliability measure for knowledge graph embeddings, in: ACM Web Conference, 2024, p. 2009–2019.

[30] L. Galárraga et al., Fast rule mining in ontological knowledge bases with amie ++, VLDBJ 24 (2015) 707–730.

[31] S. Ortona et al., Rudik: Rule discovery in knowledge bases, VLDB (2018) 1946–1949.

[32] N. Ahmadi et al., Mining expressive rules in knowledge graphs, JDIQ (2020) 1–27.