AI supported Knowledge Graph Design & Generation

Marco Ratta[000−0003−3788−6442]

The Open University, Walton Hall, Milton Keynes (UK) marco.ratta@open.ac.uk

Abstract. Knowledge Graphs (KG) have risen to be a powerful mechanism to represent knowledge. Despite this most data sources are generally still represented via heterogeneous non-graph data structures. Converting these into KGs necessitates considerable effort from experts, proving this to be a time consuming process. While tools have been developed to aid KG builders, a gap still exists in terms of technologies that support the automation of designing KG building pipelines. Addressing this gap motivates this research. To do this the aim is to first understand the problem at the knowledge level and, inspired by the recent release of generative tools such as *GPT-Engineer*, to put forward a symbolic-generative hybrid artifact aimed at solving it. We report on the preliminary findings that we have so far reached during the first year of research in deriving the requirements for building KG generating pipelines from the literature.

Keywords: Knowledge Graph Generation · Knowledge Graph Design · Artificial Intelligence · Knowledge Level · Knowledge Engineering

1 Introduction and Motivation

Knowledge Graphs [13] have risen to be one the most powerful mechanisms to represent knowledge and have been shown to play an important role for many applications, including conversationl agents, data integration, and recommender systems [32].

However, most data sources come in non-heterogeneous non-graph data structures, schemes, and formats [7]. The conversion of these sources into KG representations involves a range of specific tasks. These span from mapping tasks, such as mapping a specific data type to its specific URI format, through to ontology engineering tasks such as aligning data specific vocabulary with known vocabularies. A central problem is also the overarching organisation of these tasks into a pipeline whereby one is able to go from the original data to the KG. This whole process necessitates a considerable effort from both KG engineers and domain experts, showing this to be an often time consuming and complicated process [3].

While advances have been made in terms of supporting KG engineers in their work, "existing techniques offer [...] solutions, each covering a specific aspect of KG construction, but automatically orchestrating [...] the whole construction process remains the challenge" [14]. Addressing this challenge would pose questions such as, for example, given data sources and a target ontology can a pipeline from the sources to the target KG be generated automatically? If human intervention is required in the process, what is the nature and extent of this input? And What is the role and extent of background knowledge? Addressing this challenge and such questions is the motivation behind this research.

To accomplish this, we first aim to understand the problem of KG design and generation at the knowledge level, beginning with a review of the relevant literature looking at how KG builders have concretely generated KG building pipelines as a source of knowledge. The purpose is to discover and represent the required KG tasks and operations, and from these to build a systematising model of KG tasks and ways to compose them inspired by problem solving methods. Influenced by recent generative tools such as *GPT-Engineer*[1], where the user interacts conversationally with the agent to develop code, and under the assumption that we can model the problem of integrating structured and semi-structured sources as one of graph-to-graph transformations, the aim is to build a symbolic-generative hybrid agent that supports users in the designing and executing of KG generation pipelines for structured and semi-structured sources.

Thus, our objectives are: (i) enriching the theoretical understanding of KG design and generation by looking into the existence of abstractable KG building tasks (i.e. that go beyond a single use-case) and using these to formulate abstract high-level tasks, (ii) extending the application domain of AI to the design of KG generating pipelines through the orchestration of tasks and (iii) supporting the community of practitioners by releasing novel methods for designing and organising KG building.

The remainder of the paper is structured as follows. First, we delve into the state of the art and related work, then we proceed to state our research questions and contribution. Following from that we proceed to describe our intended methodology and the evaluation approach. Finally we report on the early findings that we have so far achieved in the abstraction of general KG construction tasks from the literature during this first year of research.

2 State of the Art

We now consider related work on knowledge engineering, data integration, KGs, KG construction, graph-to-graph transformation, ontology alignment and AI.

As it is not assumed that the required tasks and the methods for solving them are known beforehand, we take a knowledge engineering approach to tackling our problem. This entails that the practice of KG engineers and creators is taken as "a source of knowledge that cannot be obtained from anywhere else" [23] and that a knowledge based system "is not a container filled with knowledge extracted from an expert, but an operational model that exhibits some desired

[1] https://github.com/gpt-engineer-org/gpt-engineer

behaviour observed or specified in terms of real-world phenomena" [30]. Central to a knowledge engineering approach are also problem solving methods (PSMs) [9], which are domain-independent reasoning components specifying reusable patterns of behaviour. Furthermore, when we speak of automation we usually refer to the automation of a process at the knowledge level, where we consider a solution to the problem via the design of actions, goals and body of a hybrid symbolic-generative agent.

The application domain of this research falls under the umbrella of data integration processes. Lenzerini [19] formalises such a process. This combines data from heterogeneous sources into a single unified view available to a user or client. Its basic components are the heterogeneous data sources, a global schema acting as a mediator and view of this underlying data and the mappings that connect the information contained in the sources to the schema of the global view. For us this global view is a KG.

By KG we understand a graph intended to accumulate and convey knowledge of an object whose nodes represent entities of interest and whose edges represent relations between these entities [13]. A number of approaches and tools have been developed to aid KG engineers in their work. Central to the task of KG construction has been the effort put into the development of W3C recommended R2RML/RML mapping languages for the creation of RDF mappings [29][8][16] and the tools that materialise them [15][7][1][2]. Further technologies include methods for resolving data heterogeneity such as Ontology Based Data Access (OBDA) [5][4], Extract Transform Load (ETL) based tools [27], and SPARQL based tools [18]. Further, there are also multi-agency [22] and deep-learning [31] approaches for schema matching and ontology mapping, and a variety of fuzzy logic, probabilistic soft logic and machine/deep learning methods [14].

One of the foundational assumptions of this work is that we can frame the problem of designing KG generation pipelines as one of graph-to-graph manipulation. The *Facade-X* methodology [3] is therefore one of the starting points of the artifact development process as it can resolve from the start the heterogeneity of the data sources into graphs. This methodology assumes an iterative process of two phases: (a) re-engineering, where the task is to resolve the heterogeneous formats of the sources into a graph, and (b) re-modelling, where the main objective is to add semantics [6]. High-level tasks undertaken during these phases include: (i) specifying a representative ontology for the sources, (ii) extracting the information (e.g. entity recognition/resolution, schema matching, defining mappings, schema learning), (iii) generating the graph from the mappings and (iv) KG evaluation and quality assessment (completeness, redundancy, correctness, etc.). Crucially, his assumption allows to connect our approach and methods with those from the field of graph transformation/re-writing, which focuses on geometrical and rule based methods for graph to graph manipulations [10][11], and with those from ontology matching and alignment [25][17].

Recently a lot of the attention in the AI community and beyond has focused on deep learning and especially large language models (LLMs) as possibly universal means for the development of AI and real-world programming tasks [28] although not without debate in the community [21]. Their use for the generation of KGs has also been the object of engineering exploration [20][24].

Finally, the aim is to implement a methodology based on the principles of design science [26]. Design science based development is an iterative process that can be viewed as embodiment of three closely related cycles of activities that are happening simultaneously: the relevance cycle, the rigor cycle and the design cycle [12]. The relevance cycle is concerned with the introduction of input requirements from the contextual environment into the development process and introduces the research artifacts into environmental field testing. The purpose of the rigor cycle is to ensure the proper theoretical grounding of our development work. The purpose of the design cycle is instead to generate design alternatives and evaluate these alternatives against the requirements provided by the relevance cycle and the rigor cycle.

3 Problem Statement and Contributions

The overarching research question of the project is the following:

(RQ1) How can the development of Knowledge Graph generating processes from structured and semi-structured data sources be automated?

The answer to this question can be broken into two interconnected parts, each with their own sub-questions. For the first part, concerned with requirements and theoretical rigor, we ask:

(RQ2) are there general features, specifically tasks, in Knowledge Graph generation pipelines that are shared and that can be abstracted?

(RQ3) how can we systematise our findings into a model of the process of designing Knowledge Graph generating pipelines at the knowledge level?

For the second part, concerned with implementation, our target is that given a set of data sources and a set of ontologies, the system aids in the designing of a KG generating pipeline derived from the given inputs. Our assumption is that this can be achieved through a hybrid symbolic and generative AI system, supported by breaking down the problem into tasks that can be addressed by PSMs that are to be applied to graph-to-graph transformations. But is this actually required and is it beneficial with respect to an approach based say solely on using an LLM and prompt engineering? Therefore,

(RQ4) Does the breaking down of the problem into tasks and methods help with the creation of KG generating pipelines?

(RQ5) Can the process of converting and integrating structured and semi-structured sources be adequately modelled as a problem of graph-to-graph transformations?

It is also the case that, for example, in a task of entity alignment where the same entity is referenced by name in one source and by an identification number in a second sources requires a background knowledge that only a user acquainted with the sources can provide. This is a central issue as no AI agent, symbolic or not, can predict what it has not seen. Thus, it can be asked

(RQ6) how much of the process of KG design and generation can be automated?

(RQ7) Which parts of the process require necessary human intervention (e.g. providing background knowledge)? What is the nature and extent of this intervention?

(RQ8) What is the impact of this intervention on the generated pipelines? Can these be linear from source to target or must they include iterative problem solving structures?

4 Research Methodology and Approach

To carry out the research programme we intend to implement a methodology based on the principles of design science as follows.

The first step is to conduct a literature review of relevant sources. As parts of the relevance cycle and the rigor cycle are concerned with introducing requirements into the development process, the plan is to collect these empirically by looking at how KG builders have operated concretely. The target is then twofold. We look first for the KG task required to answer the research questions of the first part (RQ1-3) and we collect a variety of other data related to concrete KG pipelines. This includes for example the provenance of the sources (i.e. numerical, textual, etc.), their formats (csv, xml, etc.), the producing stakeholders of the graphs (semantic web researchers, government institutions, etc.) or the tools that have been used to generate the graphs. As this PhD is in its first year this step is being carried out at the moment.

The second step is to classify and systematise the tasks abstracted in the first to build the knowledge level model that will contain the answers to the research questions of the first part of the PhD (RQ1-3). Following this is the development on the basis of this model of a series of problem solving methods. These initial three steps are required in order to complete the theoretical background of the rigor cycle.

Once the requirements have been formalised, the next step is to begin to implement the design cycle and the application of the problem solving methods that have been developed to graph-to-graph transformations. Within this a number of crucial design choices will also require implementation. These include, for example, determining the distribution of the tasks between the artifact and the user (RQ6), the specification of the role of the user interacting with the system and the necessary background knowledge (RQ7-8) and the task distribution model between symbolic and generative AI methods.

Finally, to conclude the relevance and design cycle, we plan to implement an evaluation of the artifact produced through a compare and contrast approach, using a dataset of KGs that have been built with already tried and tested methods and thus also evaluating some of our initial assumptions (RQ4-5).

5 Evaluation Plan

We plan to evaluate our work through the following key and overarching KG construction goals, listed in an increasing order of difficulty:

1. Given the data sources and the target ontology, generate the KG building pipeline.
2. Given the data sources and multiple ontologies, select the subset of representative ontologies and generate the KG building pipeline.
3. Given applicable tasks and data sources, build a KG graph generating the relevant semantics.

To do this we plan to compare the results obtained with our artifact with those from a dataset of graphs already obtained via established KG construction methods using data from *Open Knowledge Graph* [2], healthcare data from *Data Commons* [3], cultural heritage data from the Tate gallery [4] and finally musical data from the *Neuma* [5] digital library and the *Digital and Cognitive Musicology Lab*[6].

Following FAIR [7] open science practices, we release the artifact on the *GitHub* platform to make it available for reproducibility.

6 Preliminary Findings

We now report on the early findings that we have obtained from the literature review. This is being conducted on a selection of papers from four publications, the *Semantic Web Journal* [8], the *Journal of Web Semantics* [9] and the conference proceedings from the *Extended Semantic Web Conference* and the *International Semantic Web Conference*. Isolating a small sample of relevant papers deemed relevant, we constructed a keyword search to select 71 papers has been produced for screening. Through a screening process targeting papers that describe an actual KG construction pipeline built from any source, 69 of them have been selected for examination.

After reviewing about 25% of the collected papers therefore, we have abstracted the following.

T1: URI design The task of constructing an expression pattern for a URI.
T2: URI mapping The task of mapping of a elements type or class to its specific URI expression pattern.

---
[2] https://data.open.ac.uk/
[3] https://www.datacommons.org/
[4] https://github.com/tategallery/collection
[5] http://neuma.huma-num.fr/
[6] https://github.com/DCMLab
[7] https://www.go-fair.org/fair-principles/
[8] https://www.semantic-web-journal.net/
[9] https://www.sciencedirect.com/journal/journal-of-web-semantics

T3: Prefix mapping The task of mapping of a URI to its prefix appearing in the final graph.
T4: Type mapping The task of mapping entity types in the sources to the corresponding target data type of the chosen ontology.
T5: Entity linking The task of mapping of an entity/object in the source to the corresponding entity in an existing taxonomy/graph.
T6: Entity resolution The task of working out whether multiple object references refer to the same object.
T7: Domain ontology selection The task of selecting ontologies that are relevant to the domain.
T8: Ontology feature selection The task of selecting the parts of an ontology that are descriptively relevant.
T9: Ontology composition The task of assembling together a plurality ontology parts into a single domain model.
T10: Ontological requirements specification The task of specifying the ontological requirements for representing the underlying sources.
T11: Database interlinking The task of generating triples that link a RDF database to another RDF database.
T12: Ontology extension The task of extending an ontology via the operations of subclassing and subtyping.
T13: Schema validation The task of validating that the content of the final graph conforms to the given schema or ontology.
T14: Content validation The task of validating the semantic content of the final graph.
T15: Syntax validation The task of validating the syntactic content of the final graph.
T16: Output graph syntax selection The task of selecting the syntax in which to express the final Knowledge Graph.
T17: Source to RDF mapping The task of mapping source components and/or data types to their respective RDF terms. Related to T5.
T18: Language tagging The task of adding language tags to literals.
T19: Links explicitation The task of rendering implicit links in the sources explicit by constructing appropriate triples (e.g. changing literals into resources).
T20: Subclassing The task of mapping a source class to an ontology class via rdfs:subClassOf. Related to T12.
T21: Subpropertying The task of mapping a source property to an ontology property via rdfs:subPropertyOf. Related to T12.
T22: Ontology alignment The task of linking a class or property of the ontology to a class and/or property of another ontology via a specific triple.
T23: Internal linking The task of linking unconnected resources in the source data with one another via a triple.
T24: External linking The task of linking a resource in a RDF dataset to its representation in an external dataset (i.e. Dbpedia). Related to T5.
T25: Data format modification The task of modifying the format of a piece of data in the sources to another one in the final graph.

T26: Data model Transformation The task of mapping RDF data described in one data model to a different data model.

It must be noted that the literature review is still ongoing and therefore the below list does not intend to be exhaustive. Once finished, as anticipated in section 4 above, the next step is to analyse them and systematise them into a knowledge level model, something that we have however already started doing. This includes categorising them in terms of thematic groups, identifying which tasks can be considered "atomic" and which ones can be constructed through a composition of these "atomic" tasks, determining an initial task distribution between generative and symbolic methods and aligning the results with established ones from the field of ontology matching/alignment.

7 Conclusion

In this work I have proceeded to expose my early stage (first year) PhD research on the yet open problem of theorising and applying the methods and tools of artificial intelligence to the problem of automating the designing KG generating pipelines.

To conclude, a brief discussion of some of the challenges that may be faced is required. A risk with what has been proposed is the possibly vast scale of the endeavour, the mitigation of which could lead to require some scope narrowing as we will begin to attempt to move from the knowledge level representation of our object to the building of its implementation. The early stage nature of this project also still leaves a number of open questions. A central one that may affect the final results and related to the assumptions that have been taken is the extent to which automation can be achieved. That is, how many of the overall tasks to be performed can be safely allocated to an agent and how many to the user, either by design or by necessity, cannot be answered at this stage. Providing an answer to these questions will indeed play a part in the future development of this PhD project.

Acknowledgements

This research was supported by the UK EPSRC Doctoral Training Partnership 2022-2024 Open University and partly by the EU-funded project Polifonia: a digital harmoniser of musical cultural heritage (Grant Agreement N. 101004746), https://polifonia-project.eu.

Supervisors: Dr. Enrico Daga and Dr. Paul Mulholland.

References

1. Arenas-Guerrero, J., Scrocca, M., Iglesias Molina, A., Toledo, J., Pozo-Gilo, L., Dona, D., Corcho, O., Chaves-Fraga, D.: Knowledge graph construction with r2rml and rml: an etl system-based overview. In: CEUR workshop proceedings. vol. 2873. CEUR Workshop Proceedings (2021)

2. Arenas-Guerrero, J., Chaves-Fraga, D., Toledo, J., Pérez, M.S., Corcho, O.: Morph-KGC: Scalable knowledge graph materialization with mapping partitions. Semantic Web **15**(1), 1–20 (2024). https://doi.org/10.3233/SW-223135
3. Asprino, L., Daga, E., Gangemi, A., Mulholland, P.: Knowledge graph construction with a façade: a unified method to access heterogeneous data sources on the web. ACM Transactions on Internet Technology (2022)
4. Buron, M., Goasdoué, F., Manolescu, I., Mugnier, M.L.: Obi-wan: ontology-based rdf integration of heterogeneous data. Proceedings of the VLDB Endowment **13**(12), 2933–2936 (2020)
5. Corcho, O., Priyatna, F., Chaves-Fraga, D.: Towards a new generation of ontology based data access. Semantic Web **11**(1), 153–160 (2020)
6. Daga, E., Asprino, L., Mulholland, P., Gangemi, A.: Facade-x: an opinionated approach to sparql anything. Studies on the Semantic Web **53**, 58–73 (2021)
7. Dimou, A., Chaves-Fraga, D.: Declarative description of knowledge graphs construction automation: Status & challenges. In: Proceedings of the 3rd International Workshop on Knowledge Graph Construction (KGCW 2022) co-located with 19th Extended Semantic Web Conference (ESWC 2022). vol. 3141 (2022)
8. 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. Ldow **1184** (2014)
9. Fensel, D., Motta, E.: Structured development of problem solving methods. IEEE Transactions on Knowledge and Data Engineering **13**(6), 913–932 (2001)
10. Heckel, R.: Graph transformation in a nutshell. Electronic notes in theoretical computer science **148**(1), 187–198 (2006)
11. Heckel, R., Taentzer, G.: Graph transformation for software engineers: With applications to model-based development and domain-specific language engineering. Springer (2020)
12. Hevner, A., Chatterjee, S., Hevner, A., Chatterjee, S.: Design science research in information systems. Design research in information systems: theory and practice pp. 9–22 (2010)
13. 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. ACM Computing Surveys (CSUR) **54**(4), 1–37 (2021)
14. Hur, A., Janjua, N., Ahmed, M.: A survey on state-of-the-art techniques for knowledge graphs construction and challenges ahead. In: 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). pp. 99–103. IEEE (2021)
15. Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D., Vidal, M.E.: Sdm-rdfizer: An rml interpreter for the efficient creation of rdf knowledge graphs. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 3039–3046 (2020)
16. Iglesias-Molina, A., Van Assche, D., Arenas-Guerrero, J., De Meester, B., Debruyne, C., Jozashoori, S., Maria, P., Michel, F., Chaves-Fraga, D., Dimou, A.: The rml ontology: A community-driven modular redesign after a decade of experience in mapping heterogeneous data to rdf. In: International Semantic Web Conference. pp. 152–175. Springer (2023)
17. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: ECAI. vol. 242, pp. 444–449 (2012)

18. Lefrançois, M., Zimmermann, A., Bakerally, N.: A sparql extension for generating rdf from heterogeneous formats. In: The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14. pp. 35–50. Springer (2017)
19. Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 233–246 (2002)
20. Meyer, L.P., Stadler, C., Frey, J., Radtke, N., Junghanns, K., Meissner, R., Dziwis, G., Bulert, K., Martin, M.: Llm-assisted knowledge graph engineering: Experiments with chatgpt. arXiv preprint arXiv:2307.06917 (2023)
21. Mitchell, M., Krakauer, D.C.: The debate over understanding in ai's large language models. Proceedings of the National Academy of Sciences **120**(13), e2215907120 (2023)
22. Nagy, M., Vargas-Vera, M.: Towards an automatic semantic data integration: Multi-agent framework approach. Semantic web pp. 107–134 (2010)
23. Newell, A.: The knowledge level. Artificial intelligence **18**(1), 87–127 (1982)
24. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering (2024)
25. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering **25**(1), 158–176 (2011)
26. Simon, H.A.: The science of design: Creating the artificial. Design Issues pp. 67–82 (1988)
27. Sreemathy, J., Nisha, S., RM, G.P., et al.: Data integration in etl using talend. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). pp. 1444–1448. IEEE (2020)
28. Vaithilingam, P., Zhang, T., Glassman, E.L.: Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In: Chi conference on human factors in computing systems extended abstracts. pp. 1–7 (2022)
29. Van Assche, D., Delva, T., Haesendonck, G., Heyvaert, P., De Meester, B., Dimou, A.: Declarative rdf graph generation from heterogeneous (semi-) structured data: A systematic literature review. Journal of Web Semantics **75**, 100753 (2023)
30. Wielinga, B.J., Schreiber, A.T., Breuker, J.A.: Kads: A modelling approach to knowledge engineering. Knowledge acquisition **4**(1), 5–53 (1992)
31. Zhang, J., Shin, B., Choi, J.D., Ho, J.C.: Smat: An attention-based deep learning solution to the automation of schema matching. In: Advances in Databases and Information Systems: 25th European Conference, ADBIS 2021, Tartu, Estonia, August 24–26, 2021, Proceedings 25. pp. 260–274. Springer (2021)
32. Zou, X.: A survey on application of knowledge graph. In: Journal of Physics: Conference Series. vol. 1487, p. 012016. IOP Publishing (2020)