ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA241038

KGPRUNE: A Web Application to Extract Subgraphs of Interest from Wikidata with Analogical Pruning

Pierre Monnin[a,*], Cherif-Hassan Nousradine[b], Lucas Jarnac[b,c], Laurel Zuckerman[d] and Miguel Couceiro[b,e]

[a]Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France
[b]Université de Lorraine, CNRS, LORIA, Nancy, France
[c]Orange, France
[d]Independent Researcher
[e]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
ORCID (Pierre Monnin): https://orcid.org/0000-0002-2017-8426, ORCID (Lucas Jarnac): https://orcid.org/0000-0002-2819-2679, ORCID (Laurel Zuckerman): https://orcid.org/0000-0001-8630-0554, ORCID (Miguel Couceiro): https://orcid.org/0000-0003-2316-7623

Abstract. Knowledge graphs (KGs) have become ubiquitous publicly available knowledge sources, and are nowadays covering an ever increasing array of domains. However, not all knowledge represented is useful or pertaining when considering a new application or specific task. Also, due to their increasing size, handling large KGs in their entirety entails scalability issues. These two aspects asks for efficient methods to extract subgraphs of interest from existing KGs. To this aim, we introduce KGPRUNE, a Web Application that, given seed entities of interest and properties to traverse, extracts their neighboring subgraphs from Wikidata. To avoid topical drift, KGPRUNE relies on a frugal pruning algorithm based on analogical reasoning to only keep relevant neighbors while pruning irrelevant ones. The interest of KGPRUNE is illustrated by two concrete applications, namely, bootstrapping an enterprise KG and extracting knowledge related to looted artworks.

1 Introduction

Knowledge graphs (KGs) are structured representations that model the knowledge of one or several domains. Their atomic units are triples (h, p, o) that represent the existence of a relationship p between two entities h and o. With their flexibility and the knowledge they provide, KGs have become major assets that fuel various methods of artificial intelligence (*e.g.*, retrieval augmented generation for large language models [19], machine learning in general [4]) with applications in a wide array of domains (*e.g.*, search, e-commerce, social networks, life sciences [3, 9, 18]).

In the seminal spirit of the Semantic Web [1], existing KGs are often re-used for building new KGs or for supporting new tasks or applications [5, 7, 15, 22]. This is possible due to the increasing number and size of publicly available KGs[1], and in particular of large KGs covering several domains such as Wikidata. The latter is a generic KG of more than 100 million nodes[2] that supports Wikipedia [24]. Wikidata is considered as a premium source of knowledge but several issues hinder its reusage [10, 21]. First, its large size entails scalability issues when handling the graph (*e.g.*, storage, query performance). Second, not all represented knowledge is relevant to the considered tasks or applications. For example, one of the neighboring entities of `Microsoft SharePoint` is `Dating App` which may not be of interest when building a enterprise KG modeling the IT domain.

To address such issues, several authors have proposed extracting subgraphs, either manually with some early examples dating back to 1996 [22] or automatically [11, 10, 21]. In particular, we recently proposed an approach that traverses the neighborhood of seed entities provided by users, keeping relevant neighbors while pruning irrelevant ones [11]. This approach relies on analogical inference and exhibits high performance, including in transfer settings, with a drastically low number of parameters.

Building on this previous work, we propose KGPRUNE,[3] a Web Application to extract subgraphs from Wikidata given seed entities and properties of interest to the user. KGPRUNE can be used both from a browser and programmatically through an API, allowing users with various technical expertise to interact with our pruning approach. In the following, after describing the features and technical architecture of KGPRUNE, we illustrate its interest with two concrete applications: the bootstrapping of an enterprise knowledge graph, and the extraction of knowledge related to looted artworks. A video of our demonstration is available on YouTube.[4]

2 KGPRUNE: Extracting Subgraphs of Interest

The main screens of the KGPRUNE Web Application are presented in Figure 1. We describe below the main characteristics and steps for interacting with the application.

Supporting KG. We chose to build KGPRUNE upon the Wikidata KG as it is large and generic, and thus can serve as a premium source of knowledge for several domains. However, it should be noted that our approach could be applied on any KG.

* Corresponding Author. Email: pierre.monnin@inria.fr.
[1] https://lod-cloud.net/
[2] https://www.wikidata.org/wiki/Wikidata:Statistics
[3] https://kgprune.loria.fr
[4] https://youtu.be/mt5gF4ZmhGY

Input files. KGPRUNE only requires as input from the user two CSV files, as illustrated in Table 1. The file `qid_example.csv` contains QIDs identifying seed entities of interest whose neighborhood will be retrieved. Here, as an example, we consider `Microsoft SharePoint` (Q18833) and the `Java` programing language (Q251). The file `pid_example.csv` contains PIDs identifying properties of interest whose edges will be traversed. Here, we consider `instance of` (P31), `subclass of` (P279), and `part of` (P361). Note that indicating the PID of a property leads to traversing direct edges whereas indicating (-)PID leads to traversing inverse edges. Here, both direct and inverse P279 edges will be traversed. The upload screen of KGPRUNE is presented in Figure 1a.

Table 1. Example of input files for KGPRUNE. The file `qid_example.csv` contains QIDs of seed entities of interest. Their neighborhood will be retrieved by traversing edges labeled by the properties whose PIDs are specified in `pid_example.csv`.

`qid_example.csv`	`pid_example.csv`
Q18833	P31
Q251	P279
	(-)P279
	P361

Subgraph extraction. After input CSV files have been uploaded, KGPRUNE executes our traversal and pruning algorithm [11][5]. Starting from seed entities, edges labeled by the specified properties are traversed. For each neighbor, our analogical pruning model decides either to keep or prune it. Analogies are statements of the form "A is to B as C is to D", modeled as quadruples $A : B :: C : D$ such as Paris : France :: Berlin : Germany. Such quadruples capture similarities and dissimilarities between objects [16, 17]. Here, given a seed entity e_s^u specified by the user and one of its neighbors e_r^u, our model predicts whether they form an analogy with a seed entity e_s^k and one of its neighbor e_r^k for which a "keep" decision is known:

$$\underbrace{e_s^k : e_r^k}_{\text{Known "keep" decision}} :: \underbrace{e_s^u : e_r^u}_{\text{Unknown decision}}$$

This prediction relies on the pre-learned embeddings of the entities and the convolutional model for analogy detection introduced by Lim et al. [13]. With its architecture, the analogy-based model is able to capture relative similarities and dissimilarities between seed entities and their neighbors to keep or to prune, and thus is able to generalize to heterogeneous unseen entities. If our model predicts that they form an analogy, the known decision between e_s^k and e_r^k (*i.e.*, keep e_r^k) is extrapolated to e_r^u. Otherwise, e_r^u is pruned. Note that the known decisions originate from one manually annotated dataset named `dataset1` that is publicly available[6].

This process is performed iteratively on the neighborhood of kept neighbors until no more neighbors can be reached. Results are then displayed to the user (Figure 1b) who can choose to visualize the extracted subgraphs (Figure 1c) or download them as JSON or RDF to be imported into a new KG. The visualization interface allows users to explore the neighborhoods of the seed entities, and assess the pruning results. In particular, users can notice in the UI if our model wrongfully pruned a neighbor of interest to the users, and add it to the seed entities to force its consideration. This lays the path towards an iterative pruning process in which users explore pruning results and provide feedback that is leveraged in the subsequent iterations.

[5] https://github.com/Orange-OpenSource/analogical-pruning
[6] https://doi.org/10.5281/zenodo.8091584

Technical architecture. KGPRUNE relies on the technical architecture presented in Figure 2. Users can interact with the application via a Web browser or the provided API. Their subgraph extraction tasks are sent as SLURM jobs to our computing clusters where Wikidata adjacency and pre-trained embeddings, as well as our analogical pruning models are loaded and used in inference.

For learning Wikidata embeddings, we used the TransE [2] model with a dimension of 200. For the analogical model, we trained it using using `dataset1` among two manually annotated datasets publicly available[6]. We use 16 filters on the first convolutional layer and 8 filters on the second convolutional layer.

Our model achieves competitive performance compared to the main models of the state of the art, with a drastically lower number of parameters and a superior generalization capability in a transfer setting (Table 2).

Table 2. Performance of our model compared to LSTM, its main competitor. The transfer setting corresponds to training the model on `dataset1` and testing it on `dataset2`. Experiments on each dataset were performed using 5-fold cross-validation and hypertuning the number of filters. Full results are available in [11].

Model		LSTM	Path Analogy
`dataset1`	Precision	79.72 ± 5.17	80.10 ± 0.84
	Recall	76.00 ± 6.59	74.44 ± 5.28
	F1	77.43 ± 2.38	77.06 ± 2.89
	ACC	83.48 ± 3.05	83.51 ± 2.87
	# parameters	210,751	1,401
`dataset2`	Precision	78.49 ± 8.80	81.63 ± 8.27
	Recall	94.58 ± 2.96	94.90 ± 2.16
	F1	85.36 ± 4.53	87.54 ± 5.05
	ACC	78.66 ± 5.95	82.50 ± 6.07
	# parameters	210,751	251
Transfer setting	Precision	92.83	91.49
	Recall	74.73	83.39
	F1	82.80	87.25
	ACC	80.04	84.33

3 Illustrative Use Cases

To showcase the impact of KGPRUNE, we experimented on two use cases, namely, enterprise KG bootstrapping and extracting subgraphs related to looted artworks, allowing us to attest the usefulness of our tool on distinct real-world applications.

Bootstrapping an Enterprise Knowledge Graph (EKG). Building a new KG requires its bootstrapping with a high quality nucleus that can then support automatic knowledge extraction approaches from structured or unstructured data (*e.g.*, tables, texts) [14, 20, 25]. Indeed, these approaches then enrich the KG while being guided by the terms and relations the KG provides, forming a virtuous loop.

To build such a nucleus, several authors rely on Wikidata. To limit the size of the created nucleus, they select parts of the neighborhood of seed entities of interest with a distillation [21] or a pruning process [10, 11]. Their traversal of the graph focuses on the ontology hierarchy, only upward [21] or both upward and downward [10].

In [11], we proposed our pruning approach to bootstrap an EKG focused on the IT domain, traversing the ontology upward and downward starting from seed entities of interest available in the company internal glossary. The competitive performance with low complexity obtained by our approach (Table 2) illustrates its interest for this use case. With KGPRUNE, we extended our previous approach by allowing the user to define the properties to traverse, enriching the

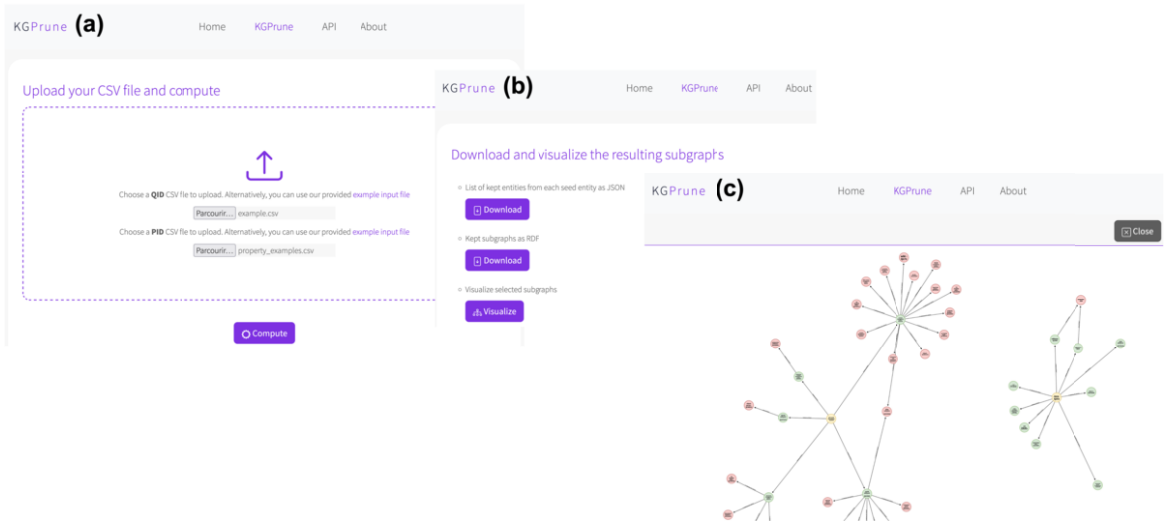

Figure 1. Main screens of KGPRUNE. (a) Upload form where two CSV files are required from the user: one indicating QIDs of seed entities and one indicating PIDs of properties to traverse. (b) Result page where the user can choose to visualize the extracted subgraphs or download them in JSON or RDF. (c) Visualization of the extracted subgraphs where seed entities are in yellow, kept neighbors in green, and pruned neighbors in red.

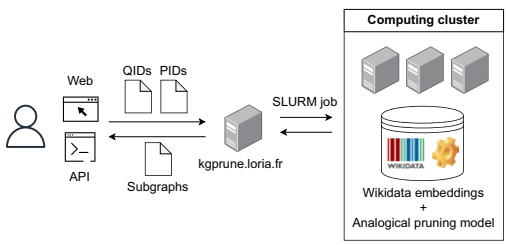

Figure 2. Technical architecture of KGPRUNE. Users can upload subgraph extraction tasks via the website or the API. These tasks are then submitted as SLURM jobs to our computing cluster.

subgraphs extracted from Wikidata, and with visualization capabilities to let user explore extracted neighborhoods.

Extracting Subgraphs Related to Looted Artworks. Art looting networks operate on many hidden levels over long periods of time. Some agencies emphasize that it is a criminal industry grossing in the billions annually. Reliable documentation is of utmost importance to finding lost or stolen cultural property, and to establishing rightful ownership. This is however a challenging task [8] since "the data on cultural heritage is locked up in data silos making it exceptionally difficult to search, locate, and obtain reliable documentation".

The authors of [8] propose the use of Linked Open Data (LOD) as a global database on cultural heritage, and explored the potential of LOD to integrate large quantities of cultural heritage data to facilitate access to information in this domain. In turn, this could help to protect cultural property from looting, as well as track looted artworks [27, 26]. However, stolen art tracking involves knowledge pertaining to artworks, genealogy, ownership, provenance, some of which is present in generic KGs together with irrelevant knowledge to the present task (*e.g.*, biology, computer science). For example, Wikidata contains 43,730 art dealers, collectors, curators, and galleries; 6,648 art museums; 930,405 paintings; 251 persons investigated by the Art Looting Investigation Unit[7]; and properties such as `owner of` and `owned by`. Hence, the need to extract specific subgraphs addressing relevent themes while avoiding false, inaccurate, or irrelevant information.

In this view, KGPRUNE has the potential to extract and collect trustworthy and pertinent information from Wikidata. In preliminary experiments, we applied our approach on the neighborhood of known artworks (*e.g.* `Cypresses`), artists (*e.g.*, `Alexej von Jawlensky`), museums (*e.g*, `National Gallery of Arts`), and art dealers (*e.g.*, `Alfred Flechtheim`). Results showed good alignment with human needs when detecting neighbors relevant to information needed for tracking stolen art. We are in the process of exploring further how KGPRUNE, and especially its pruning and visualization features, can support other use cases related to cultural heritage.

4 Conclusion and Perspectives

In this paper, we presented KGPRUNE, a Web Application allowing users to extract subgraphs of interest from Wikidata by providing seed entities of interest and properties to traverse. Our application prevents potential topical drift when traversing the graph by relying on an efficient analogy-based pruning mechanism. Users can interact with KGPRUNE via a Web browser and an API, which enables its to seamless integration in various working pipelines. We demonstrated the interest of the application with two concrete use cases.

At present, KGPRUNE only supports Wikidata. In the future, we envision to integrate additional KGs (*e.g.*, DBpedia [12], YAGO [23], Bio2RDF [6]) providing users an enhanced context from which extract subgraphs. Additionally, our analogy-based model is trained on a manually annotated dataset of seed entities and neighbors to keep or prune. Even if experiments and use cases highlighted the generalization capability of our model, it may be possible that the definition of kept and pruned neighbors learned does not apply well to other applications. To address this question, we plan on allowing users to provide their own examples of kept and pruned neighbors. These examples could be used in the inference phase or even to train tailored models on-the-fly, given the reduced complexity of our models.

[7] https://www.wikidata.org/wiki/Q30335959

Acknowledgements

This work is supported by the AT2TA project (https://at2ta.loria.fr/) funded by the French National Research Agency ("Agence Nationale de la Recherche" – ANR) under grant ANR-22-CE23-0023.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[2] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. 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*, pages 2787–2795, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[3] J. Chen, H. Dong, J. Hastings, E. Jiménez-Ruiz, V. López, P. Monnin, C. Pesquita, P. Skoda, and V. A. M. Tamma. Knowledge graphs for the life sciences: Recent developments, challenges and opportunities. *Transactions on Graph Data and Knowledge*, 1(1):5:1–5:33, 2023. doi: 10.4230/TGDK.1.1.5. URL https://doi.org/10.4230/TGDK.1.1.5.

[4] C. d'Amato, L. Mahon, P. Monnin, and G. Stamou. Machine learning and knowledge graphs: Existing gaps and future research challenges. *Transactions on Graph Data and Knowledge*, 1(1):8:1–8:35, 2023. doi: 10.4230/TGDK.1.1.8. URL https://doi.org/10.4230/TGDK.1.1.8.

[5] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610. ACM, 2014. doi: 10.1145/2623330.2623623. URL https://doi.org/10.1145/2623330.2623623.

[6] M. Dumontier, A. Callahan, J. Cruz-Toledo, P. Ansell, V. Emonet, F. Belleau, and A. Droit. Bio2rdf release 3: A larger, more connected network of linked data for the life sciences. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014*, volume 1272 of *CEUR Workshop Proceedings*, pages 401–404. CEUR-WS.org, 2014. URL https://ceur-ws.org/Vol-1272/paper_121.pdf.

[7] M. Fernández-López, A. Gomez-Perez, and N. Juristo. Methontology: from ontological art towards ontological engineering. *Engineering Workshop on Ontological Engineering (AAAI97)*, 03 1997.

[8] E. E. Fink, P. A. Szekely, and C. A. Knoblock. How linked open data can help in locating stolen or looted cultural property. In *Digital Heritage. Progress in Cultural Heritaage: Documentation, Preservation, and Protection - 5th International Conference, EuroMed 2014, Limassol, Cyprus, November 3-8, 2014. Proceedings*, volume 8740 of *Lecture Notes in Computer Science*, pages 228–237. Springer, 2014. doi: 10.1007/978-3-319-13695-0_22. URL https://doi.org/10.1007/978-3-319-13695-0_22.

[9] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers, 2021. ISBN 978-3-031-00790-3. doi: 10.2200/S01125ED1V01Y202109DSK022. URL https://doi.org/10.2200/S01125ED1V01Y202109DSK022.

[10] L. Jarnac and P. Monnin. Wikidata to bootstrap an enterprise knowledge graph: How to stay on topic? In *Proceedings of the 3rd Wikidata Workshop 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022*, volume 3262 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3262/paper16.pdf.

[11] L. Jarnac, M. Couceiro, and P. Monnin. Relevant entity selection: Knowledge graph bootstrapping via zero-shot analogical pruning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 934–944. ACM, 2023. doi: 10.1145/3583780.3615030. URL https://doi.org/10.1145/3583780.3615030.

[12] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015. doi: 10.3233/SW-140134. URL https://doi.org/10.3233/SW-140134.

[13] S. Lim, H. Prade, and G. Richard. Solving word analogies: A machine learning perspective. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 15th European Conference, ECSQARU 2019, Belgrade, Serbia, September 18-20, 2019, Proceedings*, volume 11726 of *Lecture Notes in Computer Science*, pages 238–250. Springer, 2019. doi: 10.1007/978-3-030-29765-7_20. URL https://doi.org/10.1007/978-3-030-29765-7_20.

[14] J. Liu, Y. Chabot, R. Troncy, V. Huynh, T. Labbé, and P. Monnin. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, 76:100761, 2023. doi: 10.1016/J.WEBSEM.2022.100761. URL https://doi.org/10.1016/j.websem.2022.100761.

[15] F. Mahdisoltani, J. Biega, and F. M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org, 2015. URL http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

[16] L. Miclet, S. Bayoudh, and A. Delhay. Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *Journal of Artificial Intelligence Research*, 32:793–824, 2008. doi: 10.1613/jair.2519. URL https://doi.org/10.1613/jair.2519.

[17] M. Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

[18] N. F. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. Industry-scale knowledge graphs: lessons and challenges. *Communications of the ACM*, 62(8):36–43, 2019. doi: 10.1145/3331166. URL https://doi.org/10.1145/3331166.

[19] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, page 1–20, 2024. ISSN 2326-3865. doi: 10.1109/tkde.2024.3352100. URL http://dx.doi.org/10.1109/TKDE.2024.3352100.

[20] J. Sequeda and O. Lassila. *Designing and Building Enterprise Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers, 2021. doi: 10.2200/S01105ED1V01Y202105DSK020.

[21] B. Shbita, A. L. Gentile, P. Li, C. DeLuca, and G. Ren. Understanding customer requirements - an enterprise knowledge graph approach. In *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13870 of *Lecture Notes in Computer Science*, pages 625–643. Springer, 2023. doi: 10.1007/978-3-031-33455-9_37.

[22] B. Swartout, R. Patil, K. Knight, and T. Russ. Toward distributed use of large-scale ontologies. In *Proceedings of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, volume 138, page 25, 1996.

[23] T. P. Tanon, G. Weikum, and F. M. Suchanek. YAGO 4: A reason-able knowledge base. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 583–596. Springer, 2020. doi: 10.1007/978-3-030-49461-2_34. URL https://doi.org/10.1007/978-3-030-49461-2_34.

[24] D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

[25] G. Weikum, X. L. Dong, S. Razniewski, and F. M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations and Trends Databases*, 10(2-4):108–490, 2021.

[26] L. Zuckerman. Tracking looted art with graphs: A case study. URL https://api.semanticscholar.org/CorpusID:247314664.

[27] L. Zuckerman. Linked data and holocaust era art markets: Gaps and dysfunctions in the knowledge supply chain. In *Proceedings of the International Conference Collect and Connect: Archives and Collections in a Digital Age, Leiden, the Netherlands, November 23-24, 2020*, volume 2810 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org, 2020. URL https://ceur-ws.org/Vol-2810/paper2.pdf.