

YAGO 4.6: A Logically Consistent Subset of Wikidata

Fabian M. Suchanek¹, Thomas Bonald¹, Nikola Dobričić¹, Marc Jeanmougin¹ and Margherita Necchi¹

¹Telecom Paris, Institut Polytechnique de Paris, France

Abstract

Wikidata stands as the largest public general-purpose knowledge base. Yet, as a collaborative resource, it lacks a logically consistent schema. YAGO positions itself as a logically consistent subset of Wikidata – with schema constraints on instances, relations, and classes, a manually curated taxonomy, and legible relation and instance identifiers. This demo paper presents a new version of YAGO that, for the first time, quantifies and showcases not only the statements retained for YAGO, but also those that were filtered out – shedding light on the inconsistencies inherent in large-scale collaborative knowledge bases and the trade-offs of schema enforcement.

Keywords

YAGO, knowledge base, knowledge graph, Wikidata, schema

1. Introduction

Knowledge bases (KBs), or knowledge graphs, remain foundational to a wide array of applications, even in the era of large language models: They power Graph-RAG [1], serve as evaluation benchmarks [2], enable knowledge injection into language models [3, 4], function as entity catalogs for information retrieval [5, 6, 7], and support diverse natural language processing tasks [8]. While several prominent public KBs exist [9, 10, 11, 12, 13], they differ in their maintenance models, coverage, and accessibility: Freebase [12] concluded its operation in 2015, when its content was integrated into Wikidata [14]. Babel-Net [13] is actively maintained, but its free use is restricted to the research community¹. DBpedia is also under active development, but its latest curated ontology release (of 2022²) covers only approximately 5 million instances³. Wikidata, for its part, stands as the largest collaborative general-purpose KB, boasting 100 million concepts [9] and edited by a vast community of contributors. This collaborative strength, however, introduces inherent challenges: The taxonomy can exhibit structural complexities, including hierarchical inconsistencies and rule violations [15], which occasionally complicate usage even for experienced contributors [16, 17, 18]. Recent efforts to refine this taxonomy, such as combining LLMs with graph mining [19, 20], are underway but not yet fully integrated. The WikiProject Ontology Cleaning Task Force⁴ was launched in 2023 to address critical ontology issues like class disjointness and ordering, marking an ongoing commitment to quality improvement.

The YAGO knowledge base, established in 2007 [10, 21, 22, 23, 24, 15], offers a complementary approach by prioritizing logical consistency. The KB focuses on manually defined properties, strict logical constraints, legible entity identifiers, and a curated top-level taxonomy – features that quantifiably enhance data cleanliness [15]. Since 2020, YAGO is built from Wikidata and the top-level taxonomy of `schema.org`. YAGO finds diverse applications, including in entity type prediction [25], link prediction [26], benchmarking [27, 28, 29], information retrieval [30, 31, 32, 33, 34, 35], entity

ISWC 2026 Companion Volume, October 25–29, 2026, Bari, Italy

✉ fabian.suchanek@ip-paris.fr (F. M. Suchanek); thomas.bonald@ip-paris.fr (T. Bonald); nikola.dobricic@ip-paris.fr (N. Dobričić); marc.jeanmougin@telecom-paris.fr (M. Jeanmougin); margherita.necchi@ip-paris.fr (M. Necchi)

🌐 <https://suchanek.name> (F. M. Suchanek); <https://perso.telecom-paris.fr/bonald/> (T. Bonald); TBD (N. Dobričić); <https://marc.jeanmougin.fr> (M. Jeanmougin); <https://github.com/marghenecchi> (M. Necchi)

🆔 0000-0001-7189-2796 (F. M. Suchanek); 0000-0003-0468-0384 (T. Bonald)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://babelnet.org/license>, all URLs accessed June 2026

²<https://databus.dbpedia.org/dbpedia/collections/latest-core>

³<https://www.dbpedia.org/resources/ontology/>

⁴https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology/Cleaning_Task_Force

retrieval [36, 37, 38, 5, 6, 7], and dialog systems [39].

This demo paper introduces YAGO 4.6, a new version of YAGO that, for the first time, makes transparent not just the statements retained from Wikidata, but also those that were excluded – along with the specific logical reasons for their exclusion. By quantifying and analyzing these exclusions, we provide insights into the potential inconsistencies in Wikidata, which could be of help for the Wikidata community as well. A new graphical interface allows users to interact with YAGO, and to compare directly with Wikidata.

2. Design of YAGO

Schema. YAGO is based on a manually curated top-level taxonomy. In the previous version of YAGO, this top-level taxonomy was imported from schema.org, which submitted YAGO to the rather restrictive Creative Commons Attribution-ShareAlike License⁵. In the new version of YAGO, we still use the identifiers proposed by schema.org, but we defined all classes and properties manually, thus decoupling it from schema.org. Hence, we can offer YAGO under the more permissive Creative Commons Attribution License. We also improved the schema by relaxing disjointness constraints (allowing, e.g., monuments to be both a Place and a CreativeWork). Furthermore, we moved some properties higher up in the taxonomy (allowing any person to play an instrument, and not just Musicians).

Taxonomy. Each manually defined top-level class in YAGO is linked to one or more corresponding classes in the Wikidata taxonomy; only these classes, their subclasses, and their instances are incorporated into YAGO. To ensure quality, we maintain a manually curated exclusion list of “undesired subclasses” deemed less useful for general-purpose reasoning. This list specifically targets scholarly articles (which constitute roughly half of Wikidata’s instances), meta-classes, and Unicode characters. For YAGO 4.6, we expanded this exclusion set from 19 to 77 classes, primarily by identifying previously missed synonyms⁶. Conversely, we significantly broadened our coverage of desired classes, increasing the manually defined set from 39 to 55. This expansion now includes the medical domain and bio-chemical entities (Table 1).

Class	URI	Number of instances ²
Place	schema:Place	14.3M
Person	schema:Person	13.0M
Organization	schema:Organization	6.8M
Creative work	schema:CreativeWork	4.5M
Taxon	schema:Taxon	3.9M
Product	schema:Product	2.0M
Event	schema:Event	1.1M
Bio-chemical entity ¹	schema:BioChemEntity	171k
Fictional entity	yago:FictionalEntity	168k
Intangible	schema:Intangible	99k
Medical entity ¹	schema:MedicalEntity	22k

Table 1

The 11 top-level classes of YAGO. ¹New in YAGO 4.6. ²Instances of the class or any of its subclasses.

Properties. YAGO takes only the statements of a preferredRank of Wikidata, and only those with a property that has been manually mapped to a property of YAGO. The set of manually defined properties has grown from 123 to 155, which cover 212 Wikidata properties, including 40% of the top-100 Wikidata properties. Notable additions include givenName and familyName, which are represented as human-readable strings rather than Wikidata identifiers. Numerical attributes now have type-checked metric units. Additional refinements include: coalescing identical start- and end-dates into single statements;

⁵<https://schema.org/docs/terms.html>

⁶For scholarly articles, for instance, we identified Q55915575, Q4119870, Q7318358, Q10383930, Q1907875, Q591041, Q13442814,

converting time literals from seconds to days where appropriate; removing statements involving blank nodes; filtering out comments that are substrings of other comments; and resolving a large number of user-reported bugs. Months of work went into the manual verification of missing or erroneously produced statements. While the core design principles and algorithms remain consistent with YAGO 4.5 (detailed in [15]), the codebase has been substantially rewritten to improve efficiency and reusability. Still, owing to the large size of Wikidata and the necessity to parse the full Turtle files (instead of the truthy NTriples or JSON dumps, which lack the meta-data), the overall building process takes around 12 hours on a machine with 96 cores.

3. Qualitative Analysis

	YAGO 4.5	YAGO 4.6	Wikidata
Individuals	49M	39M	120M
Classes	133k	206k	4.5M ¹
Predicates	108	155	13k
Facts	132M	167M	9.3G
Type facts	53M	43M	124M
Labels	479M	332M	720M ²
Dump size	142GB	64G	902G

Table 2

Size comparison. ¹subjects of P279. ²subjects of schema:name.

Table 2 compares the size of YAGO 4.6 with its predecessor and with Wikidata. The reduction in the number of individuals is primarily attributable to the expanded exclusion of “undesirable subclasses”. The principal novelty of YAGO 4.6 is the introduction of a comprehensive logging mechanism that records not only the statements and entities retained but also those excluded and the specific reasons for their exclusion. This enables, for the first time, a quantitative and qualitative analysis of these exclusions (Table 3).

Target	Reason	Type	Number
Taxonomy	Shortcuts	Redundancy	19k
	Excluded classes	Deliberate removal	53
	Loops	Formal issue	0
Entities	Disjointness violations	Semantic issue	1594
	No valid type	Deliberate removal	81M
	Astronomical objects	Deliberate removal	861k
	No name	Deliberate removal	152k
	Uncastable literal	Deliberate removal	4.5M
Facts	Pattern check failed	Formal issue	473
	Maxcount overflow	Formal issue	847k
	Min-max overflow	Formal issue	2
	Can’t attach property to class	Formal issue	2.8M
	Shortcut type	Redundancy	595k
	Disjoint type	Semantic issue	58k
	Domain check failed	Semantic issue	1.1M
	Range check failed	Semantic issue	1.5M
	Object not in YAGO	Formal issue	6M

Table 3

Number of elements removed from Wikidata for YAGO.

Redundancy. From the taxonomy, YAGO removes a large number of “shortcuts” (i.e., subClassOf links that are redundant), such as stepparent (Q19822352) being a subclass of human (Q5) even though

it is already a subclass of parent (Q7566), which is itself a subclass of human⁷. These links may have their *raison-d'être* in Wikidata, e.g., making the taxonomy more robust to edition errors or vandalism, but they are undesired in a KB such as YAGO that strives for logical non-redundancy.

Deliberate removal. As noted, YAGO excludes a broad range of “undesirable classes” and their instances, including scholarly articles, meta-classes, and Unicode characters. Furthermore, entities lacking names, as well as astronomical objects without common names (e.g., 2MASS J00062050-1720506 (Q222065)), are removed. These entities are not “incorrect” in Wikidata; rather, they offer limited utility for YAGO’s specific application scenarios. In a similar vein, statements containing uncastable literals are discarded. This includes strings missing obligatory language tags, such as the comment “Ing-kok bûn-hak-ka” for Douglas Adams (Q42), which is of little value to applications unable to infer the language (Taiwanese Hokkien) without explicit tags.

Formal issues. From the taxonomy, YAGO removes “loops” (i.e., circular paths), such as Squalene cyclase (Q24778143) being a sub- and superclass of Beta-amyrin synthase (Q13417346). However, the large number of “undesirable classes” entailed that no loops remained to be removed. YAGO also discards older values for functional properties (i.e., an outdated number of inhabitants of a country), and then further removes “maxcount overflows”, i.e., competing values for a functional property, cutting ties arbitrarily. Again, competing values have their reason for existence in Wikidata (which represents a “plurality of perspectives” expressed by different sources⁸), but not in YAGO, which aims for logical consistency. A very small number of statements fail syntactic pattern checks for literals (ISBNs etc.) or exceed minimum and maximum values (life expectancy has to be positive etc.). Facts whose object was removed from YAGO cannot be retained. YAGO also removes a large number of statements because they concern classes, not individuals. For instance, the class Prose (Q676) has the property *isPartOf* (P17) Literature (Q8242) in Wikidata; this fact is not included in YAGO. This is a big cut of potentially useful information, justified only by the necessity to strictly separate classes and individuals so as to allow for OWL reasoning. (We study this issue for Wikidata and YAGO as well as potential solutions elsewhere [40].)

Semantic issues (Taxonomy). Finally, YAGO removes information in order to conform to the schema. First, YAGO removes *subClassOf* links that violate disjointness constraints. In such cases, manually established priorities among the top-level classes determine which class wins. The most frequent case are titles such as Alcalde-Presidente (Q16002408), which are subclasses of both person and abstract entities such as elected office (Q17279032) in Wikidata. They become persons only in YAGO, so that any instances of Alcalde-Presidente are people, and not elected offices. Other cases appear to truly confuse different entities, such as concessions in the Dutch public transport (Q2279488), which are both organizations and permits. Here, any choice can be defended, but both together are clearly undesirable for a logically consistent KB. The vast majority of these classes is removed later anyway as they have no instances. The same process happens at the instance level, removing *rdf:type* statements that point to disjoint classes.

Semantic issues (Facts). YAGO also removes statements that violate domain and range constraints. Generally, less than 5% of the facts for any given predicate are excluded. Notable exceptions include *conferredBy*, often attached to an award ceremony instead of the award itself; *temperature*, which is sometimes incorrectly attached to buildings in Japan; *telephoneCountryCode*, erroneously linked to cities in Nigeria; and *beliefSystem*, which is often applied to organizations (e.g., the Catholic Church) rather than to the belief systems themselves (e.g., Catholicism or atheism). In total, we exclude only around 1.3M out of the 103M facts present in Wikidata for the considered predicates.

We do not attempt to “fix” these issues directly in Wikidata, as we recognize that Wikidata has valid reasons for preserving them. Instead, we generate log files in RDF-star format [41] that explain the rationale for each exclusion. We hope this transparency will assist the Wikidata community in identifying and addressing cases they deem worthy of correction. Detailed statistics regarding excluded statements and their reasons are available in the new YAGO distribution and through the demo interface.

⁷These examples may become outdated at any time, but serve as illustration.

⁸<https://www.wikidata.org/wiki/Help:Statements>

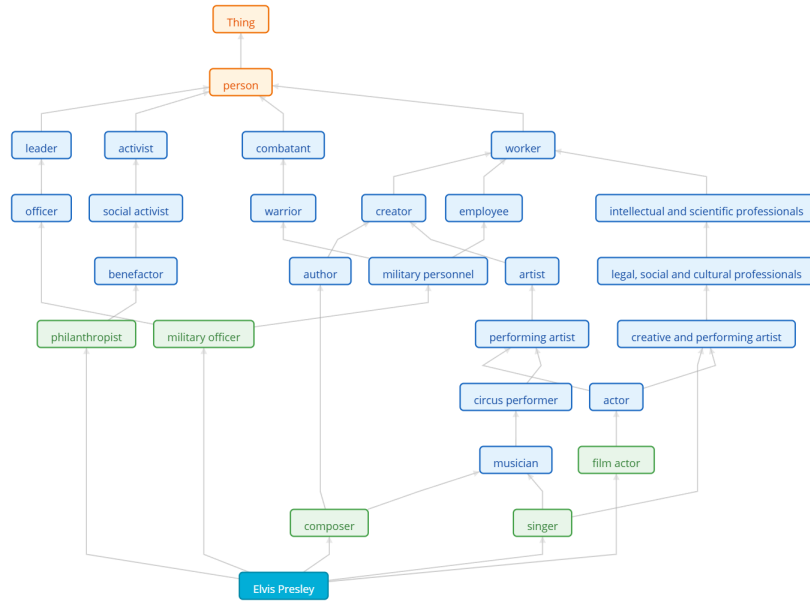


Figure 1: The taxonomy for Elvis Presley in the graphical YAGO browser.

Logical consistency. YAGO expresses constraints in the form of SHACL and OWL constraints to stipulate the presence and uniqueness of desirable properties (all individuals should have a unique label, functional properties such as isbn can have at most one value, etc.) and a correct typing of values (yago:eyeColor targets yago:Color, schema:width targets something with datatype yago:UnitOfLength such as "+1729"^^yago:Millimetre, etc.). We checked these constraints with Apache Jena [42] and pySHACL [43], and the only checking errors were negative dates (as xsd:date allows for negative dates but pySHACL does not). For reproducibility with less computational resources, we also provide a self-contained subset of YAGO ("YAGO-tiny") that encompasses the schema, the entire taxonomy, and all statements about and between a subset of 417,359 individuals.

4. Demo

The graphical interface of YAGO has been rebuilt from the ground up, with several key enhancements: First, the taxonomy is now rendered as a graph (Figure 1), with the facts displayed below. Second, the backend has been migrated to Qlever [44], improving query performance and enabling an auto-complete functionality. Most importantly, the interface now exposes, for every class and individual, the statements that were excluded during curation, along with the specific logical reasons for their removal. During the demo, users will have the opportunity to explore YAGO 4.6 and review these exclusions. The goal is to facilitate a dialogue on which statements were excluded erroneously and, conversely, which might warrant correction within Wikidata itself.

5. Conclusion

YAGO continues to position itself as a logically consistent, "cleaned" subset of Wikidata. Our demo allows users to explore not only the retained statements but also the full spectrum of excluded data. All YAGO data, the comprehensive log files detailing excluded statements, and the source code are openly available under a permissive Creative Commons Attribution (CC BY) license on the YAGO website⁹, adhering to the FAIR principles. We hope that this transparency will not only foster new applications for YAGO but also serve as a useful resource for the Wikidata community, potentially supporting efforts to improve the quality of the world's largest collaborative knowledge base.

⁹<https://yago-knowledge.org>

Acknowledgements. This work was partially supported by the Hi! PARIS and ANR France 2030 program (ANR-23-IACL-0005).

Declaration on Generative AI

Generative AI tools were used to polish the language and refine the phrasing of this manuscript.

References

- [1] T. T. Procko, O. Ochoa, Graph retrieval-augmented generation for large language models: A survey, in: 2024 Conference on AI, science, engineering, and technology (AIxSET), IEEE, 2024, pp. 166–169.
- [2] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi, When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, in: ACL, 2023, pp. 9802–9822.
- [3] OpenAI, Chatgpt plugins, <https://openai.com/blog/chatgpt-plugins>, 2023.
- [4] Q. Liu, D. Yogatama, P. Blunsom, Relational memory-augmented language models, TACL (2022).
- [5] A. Rae, V. Murdock, A. Popescu, H. Bouchard, Mining the web for points of interest, in: SIGIR, 2012.
- [6] E. R. Cinar, I. S. Altingovde, Exploiting cluster-skipping inverted index for semantic place retrieval, in: H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, B. Poblete (Eds.), SIGIR, 2023.
- [7] A. Jatowt, D. Kawai, K. Tanaka, Timestamping entities using contextual information, in: SIGIR, 2017.
- [8] G. Weikum, L. Dong, S. Razniewski, F. M. Suchanek, Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases, Foundations and Trends in Databases, 2021.
- [9] D. Vrandečić, L. Pintscher, M. Krötzsch, Wikidata: The making of, in: Companion Proceedings of the ACM Web Conference 2023, 2023.
- [10] F. M. Suchanek, G. Kasneci, G. Weikum, Yago - A Core of Semantic Knowledge , in: WWW, 2007.
- [11] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. G. Ives, Dbpedia: A nucleus for a web of open data, in: ISWC, 2007.
- [12] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: SIGMOD, 2008.
- [13] R. Navigli, M. Bevilacqua, S. Conia, D. Montagnini, F. Cecconi, Ten years of babelnet: A survey, in: IJCAI, 2021.
- [14] T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, L. Pintscher, From freebase to wikidata: The great migration, in: WWW, 2016.
- [15] F. M. Suchanek, M. Alam, T. Bonald, L. Chen, P.-H. Paris, J. Soria, YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy, in: SIGIR, 2024.
- [16] F. Brasileiro, J. P. A. Almeida, V. A. Carvalho, G. Guizzardi, Applying a Multi-Level Modeling Theory to Assess Taxonomic Hierarchies in Wikidata, in: WWW, 2016.
- [17] A. Piscopo, E. Simperl, Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata, ACM Hum. Comput. Interact. (2018).
- [18] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe, P. A. Szekely, A study of the quality of Wikidata, J. Web Semant. (2022).
- [19] Y. Peng, T. Bonald, M. Alam, Refining Wikidata taxonomy using large language models, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 5395–5399.
- [20] Y. Peng, M. Jeanmougin, T. Bonald, WiCleanData: Guaranteeing the type consistency of Wikidata by taxonomy refinement and constraint enforcement, in: Proceedings of the 35rd ACM International Conference on Information and Knowledge Management, 2026.

- [21] J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum, YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia , in: Artificial Intelligence , 2013.
- [22] J. A. Biega, E. Kuzey, F. M. Suchanek, Inside YAGO2s: A Transparent Information Extraction Architecture , in: WWW demo track, 2013.
- [23] F. Mahdisoltani, J. A. Biega, F. M. Suchanek, YAGO3: A Knowledge Base from Multilingual Wikipedias , in: CIDR, 2015.
- [24] T. P. Tanon, G. Weikum, F. M. Suchanek, YAGO 4: A Reason-able Knowledge Base , in: ESWC, 2020.
- [25] Z. Hu, V. Gutiérrez-Basulto, Z. Xiang, R. Li, J. Z. Pan, Multi-view contrastive learning for entity typing over knowledge graphs, in: EMNLP, 2023.
- [26] G. A. Gesese, R. Biswas, M. Alam, H. Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, Semantic Web 12 (2021).
- [27] C. Moon, S. Harenberg, J. Slankas, N. Samatova, Learning contextual embeddings for knowledge graph completion, in: PACIS, 2017.
- [28] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: AAAI, 2018.
- [29] S. S. Dasgupta, S. N. Ray, P. Talukdar, HYTE: Hyperplane-based temporally aware knowledge graph embedding, in: EMNLP, 2018.
- [30] R. Reinanda, E. Meij, M. de Rijke, et al., Knowledge graphs: An information retrieval perspective, Foundations and Trends in Information Retrieval 14 (2020).
- [31] M. Pasca, E. Alfonseca, Web-derived resources for web information retrieval: from conceptual hierarchies to attribute hierarchies, in: SIGIR, 2009.
- [32] E. Demidova, X. Zhou, W. Nejdl, Efficient query construction for large scale data, in: G. J. F. Jones, P. Sheridan, D. Kelly, M. de Rijke, T. Sakai (Eds.), SIGIR, 2013.
- [33] E. Baralis, L. Cagliero, S. Jabeen, A. Fiori, S. Shah, Multi-document summarization based on the yago ontology, Expert Systems with Applications 40 (2013).
- [34] G. Ercan, S. Elbassuoni, K. Hose, Retrieving textual evidence for knowledge graph facts, in: ESWC, 2019.
- [35] F. Corcoglioniti, M. Dragoni, M. Rospocher, A. P. Aprosio, Knowledge extraction for information retrieval, in: ESWC, 2016.
- [36] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum, Robust disambiguation of named entities in text, in: EMNLP, 2011.
- [37] K. Noullet, R. Mix, M. Färber, Kore 50dywc: An evaluation data set for entity linking based on dbpedia, yago, wikidata, and crunchbase, in: LREC, 2020.
- [38] L. Chen, G. Varoquaux, F. M. Suchanek, Learning high-quality and general-purpose phrase representations, arXiv preprint arXiv:2401.10407 (2024).
- [39] R. Omar, O. Mangukiya, E. Mansour, Dialogue benchmark generation from knowledge graphs with cost-effective retrieval-augmented llms, Proceedings of the ACM on Management of Data 3 (2025) 1–26.
- [40] M. Necchi, F. M. Suchanek, T. Bonald, A formal analysis of classes that act as individuals in wikidata: Punning intended, in: Wikidata Workshop at ISWC, 2026.
- [41] O. Hartig, Rdf* and sparql*: An alternative approach to annotate statements in RDF, in: ISWC poster track, 2017.
- [42] Apache Software Foundation, Apache jena, 2026. URL: <https://jena.apache.org/>.
- [43] A. Sommer, N. Car, pyshacl, 2024. URL: <https://doi.org/10.5281/zenodo.10958008>. doi:10.5281/zenodo.10958008.
- [44] H. Bast, B. Buchhold, Qlever: A query engine for efficient sparql+ text search, in: CIKM, 2017.