Using logical constraints to validate information in collaborative knowledge graphs: a study of COVID-19 on Wikidata

Houcemeddine Turki[a], Dariusz Jemielniak[b], Mohamed Ali Hadj Taieb[c], Jose Emilio Labra Gayo[d], Mohamed Ben Aouicha[e], Mus'ab Banat[f], Thomas Shafee[g], Eric Prud'Hommeaux[h], Tiago Lubiana[i], Diptanshu Das[j], Daniel Mietchen[k,*], on behalf of WikiProject COVID-19[k]

[a] Faculty of Medicine of Sfax, University of Sfax, Sfax, Tunisia
[a] turkiabdelwaheb@hotmail.fr
[b] Department of Management in Networked and Digital Societies, Kozminski University, Warsaw, Poland
[b] darekj@kozminski.edu.pl
[c,e] Faculty of Sciences of Sfax, University of Sfax, Sfax, Tunisia
[c] mohamedali.hadjtaieb@gmail.com, [f] mohamed.benaouicha@fss.usf.tn
[e] Web Semantics Oviedo (WESO) Research Group, University of Oviedo, Spain
[e] jelabra@gmail.com
[f] Faculty of Medicine, Hashemite University, Zarqa, Jordan
[f] mossab748@gmail.com
[g] La Trobe University, Melbourne, Victoria, Australia
[g] t.shafee@latrobe.edu.au
[h] World Wide Web Consortium, Cambridge, Massachusetts, United States of America
[h] eric@w3.org
[i] Computational Systems Biology Laboratory, University of São Paulo, São Paulo, Brazil
[i] tiago.lubiana.alves@usp.br
[j] Institute of Child Health (ICH), Kolkata, India
[j] Medica Superspecialty Hospital, Kolkata, India
[j] das.diptanshu@gmail.com
[k] School of Data Science, University of Virginia, Charlottesville, Virginia, United States of America
[k] dm7gn@virginia.edu

[k] Project members: Jan Ainali, Susanna Ånäs, Erica Azzellini, Mus'ab Banat, Mohamed Ben Aouicha, Diptanshu Das, Lena Denis, Rich Farmbrough, Daniel Fernández-Álvarez, Konrad Foerstner, Jose Emilio Labra Gayo, Mohamed Ali Hadj Taieb, James Hare, Alejandro González Hevia, David Hicks, Netha Hussain, Jinoy Tom Jacob, Dariusz Jemielniak, Krupal Kasyap, Will Kent, Samuel Klein, Jasper J. Koehorst, Martina Kutmon, Antoine Logean, Tiago Lubiana, Andy Mabbett, Kimberli Mäkäräinen, Bodhisattwa Mandal, Daniel Mietchen, Nandana Mihindukulasooriya, Mahir Morshed, Peter Murray-Rust, Finn Årup Nielsen, Mike Nolan, Shay Nowick, Julian Leonardo Paez, João Alexandre Peschanski, Alexander Pico, Lane Rasberry, Mairelys Lemus-Rojas, Diego Saez-Trumper, Magnus Sälgö, John Samuel, Peter J. Schaap, Jodi Schneider, Thomas Shafee, Nick Sheppard, Adam Shorland, Ranjith Siji, Michal Josef Špaček, Ralf Stephan, Andrew I. Su, Hilary Thorsen, Houcemeddine Turki, Lisa M. Verhagen, Denny Vrandečić, Andra Waagmeester, and Egon Willighagen

***Corresponding author:** Daniel Mietchen
School of Data Science, University of Virginia, Charlottesville, Virginia, United States of America
dm7gn@virginia.edu

Abstract:

Urgent global research demands real-time dissemination of precise data. Wikidata, a collaborative and openly licensed knowledge graph available in RDF format, provides a forum for exchanging structured data. In this research paper, we catalog the rules describing relational and statistical COVID-19 epidemiological data and implement them in SPARQL, a query language for semantic databases. We demonstrate the efficiency of our methods to evaluate structured information, particularly COVID-19 knowledge in Wikidata, and consequently in collaborative ontologies and knowledge graphs, and we show the advantages and drawbacks of our proposed approach by comparing it to other methods for validation of linked web data.

Keywords: SPARQL, Public health surveillance, Wikidata, Knowledge graph refinement, COVID-19, Validation constraints

1. Introduction

Since December of 2019, the COVID-19 disease has spread to become a global pandemic. This disease is caused by a zoonotic coronavirus called SARS-CoV-2 (Severe Acute Respiratory Syndrome CoronaVirus 2) and is characterized by the onset of acute pneumonia and respiratory distress. The global impact, with more than 23 million infections and 800 thousand deaths globally (as of August 25, 2020[1]), is frequently compared to the 1918 Spanish Flu (Krishnan, Ogunwole, & Cooper, 2020). As with all zoonotic diseases, its abrupt introduction to humans demands an outsized effort for data acquisition, curation and integration to drive evidence-based medicine, predictive modeling and public health policy (Dong, Du, & Gardner, 2020; Xu, Kraemer, & Group, 2020).

Agile data sharing and computer-supported reasoning about the COVID-19 pandemic and SARS-CoV-2 virus allow us to quickly learn more about the disease's epidemiology, pathogenesis, and physiopathology and to inform the required clinical, scholarly and public health measures to fight the condition and handle its non-medical ramifications (Heymann, 2020; Mietchen & Li, 2020; RDA COVID-19 Working Group, 2020). Consequently, initiatives have rapidly emerged to create datasets, web services and tools of data visualization and analysis related to COVID-19, including John Hopkins University's COVID-19 dashboard (Dong, Du, & Gardner, 2020) and the Open COVID-19 Data Curation Group's epidemiological data (Xu, Kraemer, & Group, 2020). Some of these resources are interactive and return their results based on combined clinical and epidemiological information, scholarly information and social network analysis (Cuan-Baltazar, et al., 2020; Ostaszewski, et al., 2020; Kagan, Moran-Gilad, & Fire, 2020).

Although these resources are mostly free to access, most are issued under *All Rights Reserved* terms or licenses. Similarly, 60635 (>80%) of the 74290 COVID-19-related projects on the GitHub repository for computing projects are under *All Rights Reserved*[2] terms (as of 5 August 2020), which severely impedes publication of integrated data (which ultimately undermines their value). These legal barriers block the free reuse and integration of various types of datasets and computer applications needed to achieve better analysis of COVID-19 data. There is therefore a clear need for a collaborative, free, machine-readable, and open knowledge graph integrating many varieties of information related to COVID-19 and SARS-CoV-2 for the computer-based enhancement of sustainable efforts for fighting the pandemic.

[1] "COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)". ArcGIS. Johns Hopkins University. Retrieved 25 August 2020.
[2] 60635 of 74290 as of 2020-08-05: https://github.com/search?q=covid-19+OR+covid19+OR+coronavirus+OR+cord19+OR+cord-19

Wikidata[3] fits just such a need as a CC0[4] licensed, large-scale, multilingual knowledge graph used to represent human knowledge in a structured format (Resource Description Framework or RDF) (Vrandečić & Krötzsch, 2014; Turki, et al., 2019). It therefore has the advantage of being inherently **f**indable, **a**ccessible, **i**nteroperable, and **r**eusable, i.e. FAIR (Waagmeester, et al., 2020a). It was initially developed in 2012 as an adjunct to Wikipedia but has grown significantly beyond its initial parameters. It is now a centralized, cross-disciplinary meta-database and knowledge base for storing structured information in a format optimized to be easily read and edited by both machines and humans (Erxleben, Günther, Krötzsch, Mendez, & Vrandečić, 2014). Thanks to its flexible representation of facts, Wikidata can be automatically enriched using information retrieved from multiple public domain sources or inferred from synthesised data (Turki, et al., 2019). This database includes a wide variety of pandemic-related information, including clinical knowledge, epidemiology, biomedical research, software development, geographic, demographic and genetics data and can consequently become a useful large-scale reference database to support research and medicine during the COVID-19 pandemic (Turki, et al., 2019; Waagmeester, et al., 2020a).

However, several features of projects such as Wikidata can make them at-risk of inconsistent structure or coverage: 1) collaborative projects use decentralised contribution rather than central oversight, 2) large-scale projects operate at a scale where manual checking is not possible, and 3) interdisciplinary projects draw from and integrate a wide variety of data sources. To maximise usability of the data, it is therefore important to minimise inconsistencies in its structure and coverage. As a result, methods of evaluating the existing knowledge graphs and ontologies, integral to knowledge graph maintenance and development, are of crucial importance. Such an evaluation is particularly relevant in the case of collaborative semantic databases, such as Wikidata.

Knowledge graph evaluation is a process to assess the quality, correctness, or completeness of a given knowledge graph against a set of predetermined criteria (Amith, He, Bian, Lossio-Ventura, & Tao, 2018). There are a number of possible approaches to the task, including comparing the evaluated knowledge graph to a paragon one, using the evaluated knowledge graph and judging the resulting outcomes, conducting an analysis of coverage in the evaluated knowledge graph and comparing it to the source data, and expert reviews of the evaluated ontology against certain chosen criteria (Brank, Grobelnik, & Mladenic, 2005).

Different systematic approaches have been proposed for the comparison and evaluation of ontologies and knowledge graphs, including NLP techniques, machine learning, association rule mining, and other methods (Lozano-Tello & Gomez-Perez, 2004; Degbelo, 2017; Paulheim, 2017). Ontology-based and knowledge graph-based software tools have the potential to provide data and platform interoperability, and thus, their semantic interoperability is relevant for downstream applications such as IoT and WoT technologies

[3] https://www.wikidata.org/

[4] CC0 is a rights waiver similar to Creative Commons licenses, used to publish material into the public domain. It waives as much copyright as possible within a given jurisdiction. Further information can be found at https://creativecommons.org/publicdomain/zero/1.0/

(Gyrard, Datta, & Bonnet, 2018). The criteria for evaluating ontologies typically include: *Accuracy*, which determines if definitions, classes, properties and individual entries in the evaluated ontology are correct; *Completeness*, referring to the scope of coverage of a given knowledge domain in the evaluated ontology; *Adaptability*, determining the range of different anticipated uses of the evaluated ontology (versatility); and *Clarity*, determining the effectiveness of communication of intended meanings of defined terms by the evaluated ontology (Vrandečić, 2009; Obrst, Ceusters, Mani, Ray, & Smith, 2007; Raad & Cruz, 2015; Amith, et al., 2018).

SPARQL was officially created in 2008 as a query language and protocol to search, add, modify or delete RDF data available over the Internet. Its name is a recursive acronym which stands for "SPARQL Protocol and RDF Query Language". SPARQL allows a query to be composed of triple patterns, conjunctions, disjunctions, and optional patterns and can consequently be used to retrieve contextualized information from knowledge graphs. As it has been designed to extract a searched pattern from a semantic graph (Pérez, Arenas, & Gutierrez, 2009), SPARQL queries have also been used to query the competency questions[5], so as to evaluate ontologies and knowledge graphs in a context-sensitive way (Vasanthapriyan, Tian, & Xiang, 2017; Bansal & Chawla, 2016; Martin, 2018). Indeed, a sister project presents how SPARQL can be used to generate data visualisations[6] (Nielsen, Mietchen & Willighagen 2017; Addshore, Mietchen & Willighagen, 2020). Validating RDF data portals using SPARQL queries has been regularly proposed as an approach that gives great flexibility and expressiveness (Labra Gayo & Alvarez Rodríguez, 2013). However, academic literature is still far from revealing a consensus on methods and approaches to evaluate ontologies using this query language (Walisadeera, Ginige, & Wikramanayake, 2016), and other approaches have been proposed for validation (Thornton, et al., 2019; Labra-Gayo, et al., 2019).

In this research paper, we catalogue logical constraints for the statement and dissemination of COVID-19 semantic data. We implement them with SPARQL and test them on Wikidata using the SPARQL endpoint of this knowledge graph, available at https://query.wikidata.org. We introduce the value of Wikidata as a multi-purpose collaborative knowledge graph for the flexible and reliable representation (Section 2) and validation (Section 3) of COVID-19 knowledge. We cover the use of SPARQL to query this knowledge graph (Section 4). Then, we demonstrate how logical constraints can be captured in structural schemas and consequently used to validate and encourage the consistent usage of relation types to represent COVID-19 knowledge (Section 5) and we show how statistical constraints can be applied to verify epidemiological data related to the pandemic (Section 6).

---

[5] Competency questions: A set of requirements ensuring consistency of a knowledge graph, constraints determining what knowledge to be involved in a knowledge graph (Wiśniewski, Potoniec, Ławrynowicz, & Keet, 2019).

[6] For SPARQL-based visualizations of COVID-19 information in Wikidata, see https://speed.ieee.tn/, https://egonw.github.io/SARS-CoV-2-Queries/, https://www.wikidata.org/wiki/Wikidata:WikiProject_COVID-19/Queries, and https://scholia.toolforge.org/topic/Q84263196.

Finally, we compare our constraint-based approaches with other methods (Section 7), and draw conclusions for future directions (Section 8).

2. Wikidata as a collaborative knowledge graph

Wikidata currently serves as a semantic framework for a variety of scientific initiatives, such as GeneWiki (Burgstaller-Muehlbacher, et al., 2016), allowing different teams of scholars to upload valuable academic data into a collective and standardized pool. Its versatility and interconnectedness are making it a standard for inter-disciplinary data integration and dissemination across fields as diverse as linguistics, information technology, film studies, or medicine (Turki, et al., 2019; Mitraka, et al., 2015; Mietchen, et al., 2015; Waagmeester, Schriml, & Su, 2019, Turki, Vrandečić, Hamdi, & Adel, 2017; Wasi, Sachan, & Darbari, 2020; Heftberger, Höper, Müller-Birn, & Walkowski, 2020), although its popularity and recognition across fields still vary significantly (Mora-Cantallops, et al., 2019).

It contains concepts, linked by their taxonomic relations, allowing embedding and creating instances of subclasses of classified data and links between them. Because of its multilingual nature, it is particularly useful for dynamic data reuse, as well as complex, multi-criteria data queries, and helps both in a rapid reduction of inaccuracies across Wikipedias, as well as in their fast-updating (Müller-Birn, Karran, Lehmann, & Luczak-Rösch, 2015), and generally seems to be less prone to local culture (Miquel-Ribé & Laniado, 2018) and language biases (Kaffee, et al., 2017) that are visible on Wikipedia (Jemielniak & Wilamowski, 2017).

The data structure employed by Wikidata is intended to be highly standardized, whilst maintaining the flexibility to be applied across highly diverse use-cases. There are mainly two essential components: Items, which represent objects, concepts or topics; and properties, which describe how one item relates to another. A statement, therefore, consists of a subject item (*S*), a property that describes their nature of the statement (*P*), and an object (*O*) that can be an item, a value, an external ID, or a string, etc. While items can be freely created, new properties require community discussion and vote, with 7851 properties [7] currently available. Statements can be further modified by any number of qualifiers to make them more specific and be supported by references to indicate the source of the information.

Thus, Wikidata forms a continuously growing, single, unified network graph, with 88M items forming the nodes, and 1127M statements[8] forming the edges. A live SPARQL endpoint and query service, regular RDF dumps, as well as linked data APIs and visualization tools, form a backbone of Wikidata uses (Malyshev, Krötzsch, González, Gonsior, & Bielefeldt, 2018; Nielsen, Mietchen, & Willighagen, 2017).

Importantly, Wikidata is based on free and open-source philosophy and software and is a database that anyone can edit, similarly to the very popular online encyclopedia,

[7] For an updated list of available Wikidata property, please see https://tools.wmflabs.org/hay/propbrowse/.
[8] To track the evolution of the number of Wikidata statements, please see https://grafana.wikimedia.org/d/000000182/wikidata-datamodel-references?orgId=1&refresh=30m.

Wikipedia (Jemielniak, 2014). As a result, the emerging ontologies are created entirely collaboratively, without a centralized coordination center (Piscopo & Simperl, 2018), and developed in a community-driven fashion (Samuel, 2017). This approach allows for the dynamic development of areas of interest for the user community but poses challenges, e.g., to systematic and proportionate class completeness across topics (Luggen, Difallah, Sarasua, Demartini, & Cudré-Mauroux, 2019). Also, since the edit history is available to anyone, tracing human and non-human contributions, as well as detecting and reverting vandalism is available by design and relies on peer control (Pellissier Tanon & Suchanek, 2019).

Other ontological databases and knowledge graphs exist (Färber, Bartscherer, Menne, & Rettinger, 2018; Pillai, Soon, & Haw, 2019). However, much like the factors that led Wikipedia to rise to be a dominant encyclopedia (Shafe, et al.,2017; Jemielniak & Wilamowski, 2017), Wikidata's close connection to Wikimedia volunteer communities and wide readership provided by Wikipedia have quickly given it a competitive edge. The system, therefore, aims to combine the wisdom of the crowds with advanced algorithms. For instance, Wikidata editors are assisted by a property suggesting system, proposing additional properties to be added to entries (Zangerle, Gassler, Pichl, Steinhauser, & Specht, 2016). Wikidata has subsequently exhibited the highest growth rate of any Wikimedia project and was the first amongst them to pass one billion contributions (Waagmeester, et al., 2020b).

As a collaborative venture, its governance model is similar to Wikipedia (Lanamäki & Lindman, 2018), but with some important differences. Wide permissions to edit Wikidata are manually granted to approved bots and to Wikimedia accounts that are at least 4 days old and have made at least 50 edits using manual modifications or semi-automated tools for editing Wikidata[9]. These accounts are supervised by a limited number of experienced administrators to prevent misleading editing behaviors (such as vandalism, harassment, and abuse) and to ensure a sustainable consistency of the information provided by Wikidata[10]. As such, Wikidata is highly relevant to the computer-supported collaborative work (CSCW) field, yet the number of studies of Wikidata from this perspective is still very limited (Sarasua, et al., 2019). To understand the value of using SPARQL to validate the usage of relation types in collaborative ontologies and knowledge graphs, it is important to understand the main distinctive features of Wikidata as a collaborative project.

Much as Wikidata is developed collaboratively by international editors, it is also designed to be language-neutral. As a result, it is quite possible to contribute to Wikidata with only a limited command of English and to effectively collaborate whilst sharing no common human language - an aspect unique even in the already rich ecosystem of collaborative projects (Jemielniak & Przegalinska, 2020). It may well be an early sign of other language-independent cooperative knowledge creation initiatives, such as Wikilambda, which is an abstract Wikipedia currently developed on the basis of Wikidata (Vrandečić, 2020).

---

[9] For an overview of the semi-automated editing tools for Wikidata, please see https://www.wikidata.org/wiki/Wikidata:Tools

[10] Further information about the rights and governance of users in Wikidata is shown at https://www.wikidata.org/wiki/Wikidata:User_access_levels

It is also possible to build Wikipedia articles, especially in underrepresented languages, based on Wikidata data only, and create article placeholders to stimulate encyclopedia articles' growth (Kaffee et al., 2018). This stems from combining concepts that are relatively easily intertranslatable between languages (e.g. professions, causes of death, capitals) with language-agnostic data (e.g. numbers, geographical coordinates, dates). As a result, Wikidata is a paragon example of not only cross-cultural cooperation but also human-bot collaborative efforts (Piscopo, 2018; Farda-Sarbas, et al., 2019). Given the large-scale crowdsourcing efforts in Wikidata and the use of bots and semi-automated tools to mass edit Wikidata, its current volume is higher than what can be reviewed and curated by administrators manually. It is quite intuitive: as the general number of edits created by bots grows, so grows the number of administrative tasks to be automated. Automation may include simplifying alerts, fully and semi-automated reverts, better user tracking, or automated corrections. However, the creation of automated methods for the verification and validation of the ontological relations it contains is required most.

3. Knowledge graph validation of Wikidata

As Wikidata properties are assigned labels, descriptions and aliases in multiple languages (Red in Fig. 6), multilingual information of these properties can be used alongside the labels, descriptions, and aliases of Wikidata items to verify and find sentences supporting biomedical statements in scholarly outputs (Zhang, et al., 2019). Such a process can be based on various natural language processing techniques, including word embeddings (Zhang, et al., 2019; Chen, et. al., 2020) and semantic similarity (Ben Aouicha & Hadj Taieb, 2016). These techniques are robust enough to achieve an interesting level of accuracy, and some of them can achieve better accuracy when the Wikidata classes of the subject and object of semantic relations are given as inputs (Lastra-Díaz, et al., 2019; Hadj Taieb, Zesch, & Ben Aouicha, 2020).

The subjects and objects of Wikidata relations can likewise be aligned to other biomedical semantic resources such as MeSH and UMLS Metathesaurus (Turki, et al., 2019). Thus, benchmarks for relation extraction based on one of the major biomedical ontologies can be converted into a Wikidata friendly format and used to automatically enrich Wikidata with novel biomedical relations or to automatically find statements supporting existing biomedical Wikidata relations (Zhang, et al., 2018). Furthermore, MeSH keywords of scholarly publications can be converted into their Wikidata equivalents, refined using citation and co-citation analysis (Turki, 2018), and used to verify and add biomedical Wikidata relations, e.g. by applying deep learning-based bibliometric-enhanced information retrieval techniques (Mayr, Scharnhorst, Larsen, Schaer, & Mutschke, 2014; Turki, Hadj Taieb, & Ben Aouicha, 2018).

Another option of validating biomedical statements based on the labels of their subjects, predicates, and objects in Wikidata can be the use of these labels for the reformulation of a query to search bibliographic databases and consequently to find appropriate references for the assessed Wikidata statements (Example in Fig. 5). Several bots and bot frameworks have been successfully built using this principle such as Wikidata

Integrator[11] that extracts the Wikidata statements of a given gene or protein using SPARQL, compare them with their equivalents in other structured databases like NCBI's Gene resources and Uniprot and adjust them if needed, and RefB[12] (Fig. 1) that extracts biomedical Wikidata statements not supported by references using SPARQL and identifies the sentences supporting them in scholarly publications using PubMed Central search engine and a variety of techniques such as concept proximity analysis.

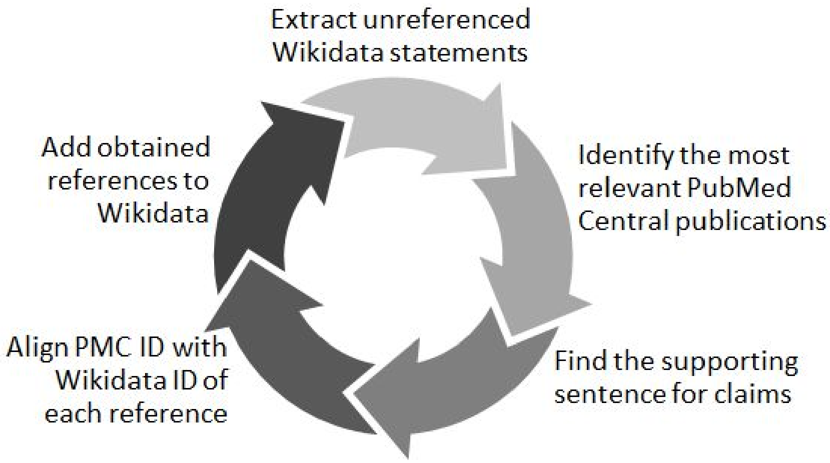

Figure 1: Process of RefB, a bot that adds scholarly references to biomedical Wikidata statements based on PubMed Central [Source: https://w.wiki/an$, License: CC BY 4.0]. The source code of RefB is available at https://github.com/Data-Engineering-and-Semantics/refb/

In addition to their multilingual set of labels and descriptions, Wikidata properties are assigned object types using wikibase:propertyType relations (Blue in Fig. 2). These relations allow the assignment of appropriate objects to statements, so that non-relational statements cannot have a Wikidata item as an object, while objects of relational statements are not allowed to have data types like a value or a URL (Vrandečić & Krötzsch, 2014).

[11] Wikidata Integrator is a bot framework for automatically curating genetic information provided by Wikidata (https://github.com/SuLab/WikidataIntegrator). For Wikidata bots using this frameworks, refer to https://www.wikidata.org/wiki/Wikidata:WikiProject_Gene_Wiki#Bot_accounts.
[12] https://www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/RefB_(WikiCred)

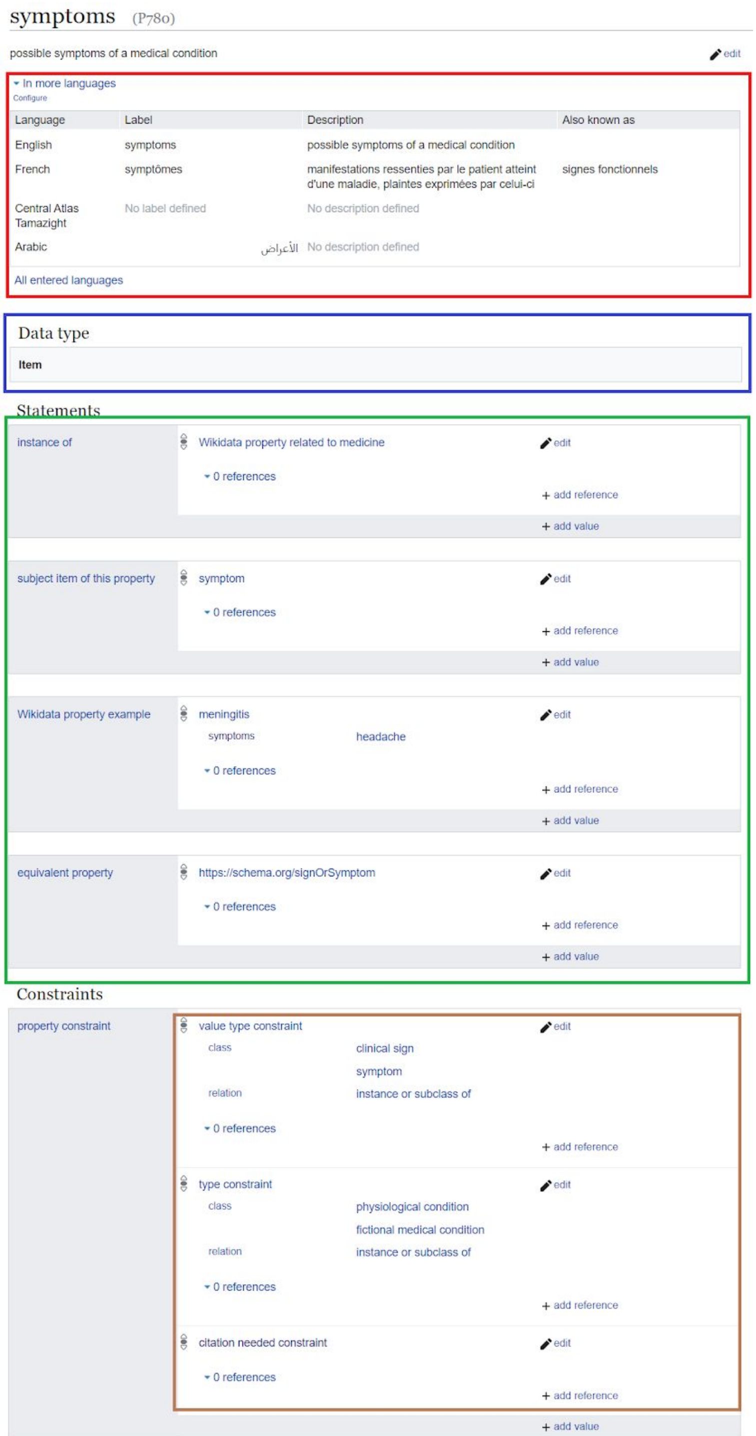

Figure 2: Wikidata page of a clinical property [Source: https://w.wiki/aeF, Derived from: https://w.wiki/aeG, License: CC BY-SA 4.0]. It includes the labels, descriptions and aliases of the property in multiple languages (Red), the object data type (Blue), statements where the property is the subject (Green) as well as property constraints (Brown).

Just like a Wikidata item, a property can be described by statements (Green in Fig. 2). These statements can link the respective property to their class (using *instance of* [P31] as a predicate), to their corresponding Wikidata item (using *subject item of this property* [P1629] as a predicate), to an example of the usage of this property (using *Wikidata property* *example* [P1855] as a predicate), to its equivalents in other IRIs[13] (using *equivalent property* [P1628] as a predicate), to the Wikimedia category that tracks its usage on a given wiki (using *property usage tracking category* [P2875] as a predicate), to its inverse property (using *inverse property* [P1696] as a predicate), or to its proposal discussion (using *property proposal discussion* [P3254] as a predicate), etc.

These statements can be interesting for various knowledge graph validation purposes. In fact, the class, the usage examples and the proposal discussion of a Wikidata property can be useful through the use of several natural language processing techniques, particularly semantic similarity, to provide several features of the use of the property such as its domain of application (e.g. the subject or object of a statement using a Wikidata property related to medicine should be a medical item) and consequently to eliminate some of erroneous use by screening the property usage tracking category. The class of the Wikidata item corresponding to the property can be used to identify the field of work of the property and thus flag some inappropriate applications. In addition, the external identifiers of such an item can be used for the verification of biomedical relations by their identification within the semantic annotations of scholarly publications built using the *SAT+R* (Subject, Action, Target, and Relations) model (Piad-Morffis, Gutiérrez, & Muñoz, 2019). The inverse property relations can identify missing Wikidata statements (C_1, P, C_2), which are implied by the presence of inverse statements (C_2, P^{-1}, C_1) in other Wikidata resources. Here, P^{-1} is the Wikidata property that is the inverse of P, C_S is a common class of the subjects of P, and C_O is a common class of the objects of P.

Despite the importance of these statements defining properties, *property constraint* [P2302] relations (Brown in Fig. 2) are the semantic relations that are primarily used for the validation of the usage of a property. In essence, they define a set of conditions for the use of a property, including several heuristics for the type and format of the subject or the object, information about the characteristics of the property, and several description logics for the usage of the property as shown in Table 1. Property constraints are either manually added by Wikidata users or inferred with an excellent accuracy from the knowledge graph of Wikidata or the history of human changes to Wikidata statements (Pellissier Tanon, Bourgaux, & Suchanek, 2019; Hanika, et al., 2019).

Table 1: Constraint types for the usage of Wikidata properties

Wikidata ID	Constraint type	Description
Q19474404	single value constraint	Constraint used to specify that this property generally contains a single value per item
Q21502404	format constraint	Constraint used to specify that the value for this property has to correspond to a given pattern
Q21502408	mandatory constraint	status of a Wikidata property constraint: indicates that the specified constraint applies to the subject property without exception and must not be violated
Q21502410	distinct values constraint	Constraint used to specify that the value for this property is likely to be different from all other items
Q21510852	Commons link constraint	Constraint used to specify that the value must link to an existing Wikimedia Commons page

[13] Internationalized Resource Identifier (IRI) is a standardized character string (e.g. a URL) that recognizes a given item in a semantic resource

Q21510854	difference within range constraint	Constraint used to specify that the value of a given statement should only differ in the given way. Use with qualifiers minimum quantity/maximum quantity
Q21510856	mandatory qualifier constraint	Constraint used to specify that the listed qualifier has to be used
Q21510862	symmetric constraint	Constraint used to specify that the referenced entity should also link back to this entity
Q21510863	used as qualifier constraint	Constraint used to specify that a property must only be used as a qualifier
Q21510864	value requires statement constraint	Constraint used to specify that the referenced item should have a statement with a given property
Q21510495	relation of type constraint	relation establishing dependency between types/metalevels of its members
Q21510851	allowed qualifiers constraint	Constraint used to specify that only the listed qualifiers should be used. Novalue disallows any qualifier
Q21510865	value type constraint	Constraint used to specify that the referenced item should be a subclass or instance of a given type
Q21514353	allowed units constraint	Constraint used to specify that only listed units may be used
Q21510857	multi-value constraint	Constraint used to specify that a property generally contains more than one value per item
Q21510859	one-of constraint	Constraint used to specify that the value for this property has to be one of a given set of items
Q21510860	range constraint	Constraint used to specify that the value must be between two given values
Q21528958	used for values only constraint	Constraint used to specify that a property can only be used as a property for values, not as a qualifier or reference
Q21528959	used as reference constraint	Constraint used to specify that a property must only be used in references or instances of citation (Q1713)
Q25796498	contemporary constraint	Constraint used to specify that the subject and the object have to coincide or coexist at some point of history
Q21502838	conflicts-with constraint	Constraint used to specify that an item must not have a given statement
Q21503247	item requires statement constraint	Constraint used to specify that an item with this statement should also have another given property
Q21503250	type constraint	Constraint used to specify that the item described by such properties should be a subclass or instance of a given type
Q54554025	citation needed constraint	Constraint specifies that a property must have at least one reference
Q62026391	suggestion constraint	status of a Wikidata property constraint: indicates that the specified constraint merely suggests additional improvements, and violations are not as severe as for regular or mandatory constraints
Q64006792	lexeme value requires lexical category constraint	Constraint used to specify that the referenced lexeme should have a given lexical category
Q42750658	value constraint	class of constraints on the value of a statement with a given property. For constraint: use specific items (e.g. "value type constraint", "value requires statement constraint", "format constraint", etc.)
Q51723761	no bounds constraint	Constraint specifies that a property must only have values that do not have bounds
Q52004125	allowed entity types constraint	Constraint used to specify that only listed entity types are valid for this property
Q52060874	single best value constraint	Constraint used to specify that this property generally contains a single “best” value per item, though other values may be included as long as the “best” value is marked with preferred rank
Q52558054	none of constraint	Constraint specifying values that should not be used for the given property

Q52712340	one-of qualifier value property constraint	Constraint used to specify which values can be used for a given qualifier when used on a specific property
Q52848401	integer constraint	Constraint used when values have to be integer only
Q53869507	property scope constraint	Constraint to define the scope of the property (main value, qualifier, references, or combination); only supported by KrBot currently

As shown in Fig. 2, a property constraint is defined as a relation where the property type is featured as an object and the detailed conditions of the constraint to be applied on Wikidata statements are integrated as qualifiers to the relation. When a property constraint is violated, the corresponding statement is automatically included in a report of property constraint violations[14] and is marked by an exclamation mark on the page of the subject item (Fig. 3) so that it can be quickly processed and adjusted by the community or by Wikidata bots if applicable.

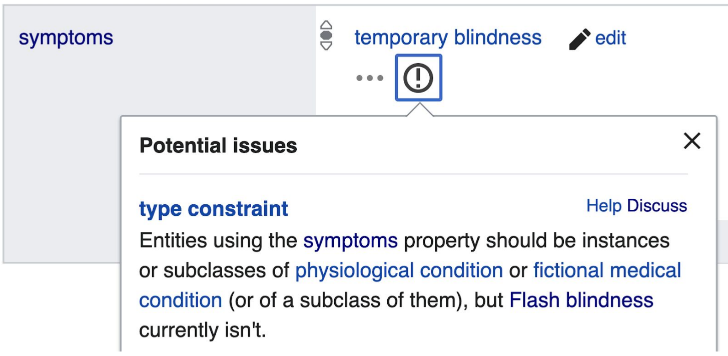

Figure 3: Example of a property constraint violation marked in the page of a Wikidata item, Q3603152 (flash blindness) [Available on Wikimedia Commons: https://w.wiki/ZuJ, license: CC0]

Although these methods are important to verify and validate Wikidata, they are not the only ones that are used for these purposes. In 2019, Wikidata announced the adoption of Shape Expressions language (ShEx) as part of the Mediawiki entity schemas extension[15]. ShEx was proposed following an RDF validation workshop that was organized by W3C[16] in 2014 as a concise, high-level language to describe and validate RDF data (Prud'hommeaux, Labra Gayo, & Solbrig, 2014). This Mediawiki extension uses ShEx to store structure definitions (EntitySchemas or Shapes) for sets of Wikidata entities which are selected by some query pattern (frequently the involvement of said entities in a Wikidata class). This provides collaborative quality control where the community can iteratively develop a schema and refine the data to conform to that schema. For those familiar with XML, ShEx is analogous to XML Schema or RelaxNG. *SHACL* (Shapes Constraint Language), another language used to constraint RDF data models, uses a flat list of constraints, analogous to XML's Schematron. It was adapted from SPIN (SPARQL Inference Notation) by the W3C Data Shapes working group in 2014 and became a W3C recommendation in 2017 (Knublauch & Kontokostas, 2017). However, ShEx has was chosen to represent EntitySchemas in Wikidata, as it has a compact syntax which makes it more human-friendly, supports recursion, and is designed to support distributed networks of reusable schemas (Labra Gayo, Prud'hommeaux, Boneva, &

[14] https://www.wikidata.org/wiki/Wikidata:Database_reports/Constraint_violations
[15] https://www.mediawiki.org/wiki/Extension:EntitySchema
[16] https://www.w3.org/2012/12/rdf-val/report

Kontokostas, 2017). Besides the possibility to infer ShEx expressions from the screening of a large set of concerned items, they can be easily written by humans in an intuitive way.

In Wikidata, ShEx-based EntitySchemas are assigned an identifier (a number beginning with an E) as well as labels, descriptions, and aliases in multiple languages, so that they can be easily identified by users. Entity schemas are defined using the ShEx-compact syntax[17], which is a concise,human-readable syntax. A schema usually begins by some prefix declarations similar to SPARQL. An optional start definition declares the shape which will be used by default. In the example (Fig. 4), the shape `<app>` will be used, and its declaration contains a list of properties, possible values, and cardinalities. By default, shapes are open, which means that other properties apart from the ones declared are allowed. In this example, the values of property `wdt:P31` are declared to be either a COVID-19 dashboard (`wd:Q90790055`), a search engine (`wd:Q91136116`) or a dataset (`wd:Q91137337`). The `EXTRA` directive indicates that there can be additional values for property `wdt:P31` that differ from the specified ones. The value for property `wdt:P1476` is declared to be zero or more literals. The cardinality indicators come from regular expressions, where '?' means zero or one, '*'; means zero or more, and '+' means one or more. While the values for the other properties are declared to be anything (the dot indicates no constraint) zero or more times, except for the properties `wdt:P577` and `wdt:P7103` that are marked as optional using the question mark. Further documentation about ShEx can be found at http://shex.io/ and in Labra Gayo et al. (2017).

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>

start = @<app>

<app>  EXTRA wdt:P31  {
  wdt:P31    [ wd:Q90790055 # instance of COVID-19 dashboard or
               wd:Q91136116 # search engine or
               wd:Q91137337 # dataset
             ] ;
  wdt:P1476 LITERAL * ; #title
  wdt:P366  .       * ; #use
  wdt:P123  .       * ; #publisher
  wdt:P178  .       * ; #developers
  wdt:P495  .       * ; #country of origin
  wdt:P306  .       * ; #operating system
  wdt:P856  .       * ; #official website
  wdt:P921  .       * ; #main subject
  wdt:P144  .       * ; #based on
  wdt:P577  .       ? ; #publication date
  wdt:P7103 .       ? ; #start of covered period
  wdt:P275 .        * ; #copyright license
  wdt:P5008 .       * ; #on focus list of Wikimedia project
}
```

[17] ShEx schemas can also be defined in RDF-based representations like Turtle or JSON-LD.

Figure 4: EntitySchema for COVID-19 dashboards, search engines and datasets [Source: https://www.wikidata.org/wiki/EntitySchema:E205]

Due to the ease of using ShEx to define EntitySchemas, it has been used successfully to validate Danish lexemes in Wikidata (Nielsen, Thornton, & Labra-Gayo, 2019) and biomedical Wikidata statements (Thornton, et al., 2019). During the COVID-19 pandemic, Wikidata's data model of every COVID-19-related class as well as of all major biomedical classes has been converted to an EntitySchema, so that it can be used to validate the representation of COVID-19 Wikidata statements (Waagmeester, et al., 2020a). These EntitySchemas were successfully used to enhance the development and the robustness of the semantic structure of the data model underlying the COVID-19 knowledge graph in Wikidata and are accordingly made available at a subpage of Wikidata's WikiProject COVID-19, accessible via https://www.wikidata.org/wiki/Wikidata:WikiProject_COVID-19/Data_models.

4. SPARQL as a semantic query language

Like ShEx, SPARQL[18] is a human-friendly language based on defining triples as conditions (Pérez, Arenas, & Gutierrez, 2009). Just like ShEx, SPARQL defines prefixes to abbreviate IRIs (Blue in Fig. 4). It also uses the skeleton of SQL to define queries to knowledge graphs in RDF format (Kumar, Kumar & Kumar, 2011). Quite similarly to SQL, SPARQL uses SELECT clauses to define the variables to show, SELECT DISTINCT clauses to define variables and prevent repeated results, FROM clauses to define the source database of the defined query, WITH clauses to define a subquery, WHERE clauses to state conditions in the form of triples, LIMIT clauses to restrict the number of returned results to a given value, OFFSET clauses to skip a number of first results, HAVING clauses to define logical conditions for filtering the variables based on aggregate functions, GROUP BY clauses to group entries to compute an aggregate function, ORDER BY clauses to sort the results according to a given variable (Kumar, Kumar, & Kumar, 2011; Bonifati, Martens, & Timm, 2017). Most of the aggregate functions used to compute new variables based on the ones retrieved by the query from the source database exist in both languages, as shown in Table 2 (Bonifati, Martens, & Timm, 2017; DuCharme, 2013). In SQL as well as SPARQL language, new aggregate function-based variables are defined in the SELECT clause using the (function(variable) AS new_variable) format, and constant values and strings are put between quotation marks.

Table 2: List of aggregate functions available in SQL and SPARQL (Bonifati, Martens, & Timm, 2017; DuCharme, 2013)

Function	Description
AVG	Average of non-NULL values in a set.
COUNT	Number of results in a group, including the ones with NULL values
MAX	Maximum in a set of non-NULL values
MIN	Minimum in a set of non-NULL values
STDEV	Standard deviation of all values provided in the expression based on a limit set of results
STDEVP	Standard deviation for all values in the provided expression based on all the returned results
SUM	Sum of all non-NULL values in a set

[18] An open license SPARQL textbook available in multiple languages can be found at https://en.wikibooks.org/wiki/SPARQL.

VAR	Statistical variance of values in an expression based on a limit set of results
VARP	Statistical variance of values in an expression based on all the returned results

Furthermore, both SQL and SPARQL define logical conditions in the HAVING clause for variables based on aggregate functions or in the WHERE clause for variables to be retrieved from the source database as FILTER (condition) (Bonifati, Martens, & Timm, 2017; DuCharme, 2013). The declaration of the logical conditions also uses the same operators (AND [also &&], || [OR], and NOT [also !]), values (True, False, and Null), logical functions (EXISTS [verifies the existence of a condition or a statement], NOT EXISTS [the opposite of EXISTS], and MINUS [eliminate the set of values having a given characteristic from the results]), and mathematical operators (> [superior to], < [inferior to], = [equal to], >= [superior or equal to], <= [inferior or equal to], != [different from], + [plus], - [minus], * [times], and / [divide]) (DuCharme, 2013).

In contrast to SQL, the variables in SPARQL are preceded by an interrogation mark and are not separated by a comma in the SELECT clause (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013), and even the declaration of statements in the WHERE clause using SPARQL is different from the one using SQL. In the latter, the declaration of the statements in a WHERE clause can only be done in a single line (Kumar, Kumar & Kumar, 2011). When multiple statement conditions should be fulfilled, they have to be linked using the INTERSECT operator (Hsu & Parker, 1995). When a unique condition from a list of statements should be respected, the list's statements should be linked using the UNION operator (Hsu & Parker, 1995). Where results fulfilling a given condition should be eliminated, the condition must be preceded by the MINUS operator (Hsu & Parker, 1995). In SPARQL, the WHERE clause can include multiple lines between curly brackets, where each line is in the form of a subject-predicate-object triple (Kumar, Kumar, & Kumar, 2011). When the statements between brackets are in the form of a triple, they should end with a period. When two successive statements have the same subject, the first statement can end with a semicolon. In this particular situation, the subject of the second statement can be omitted (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013). An exception to this is the FILTER() function allowing the definition of a logical condition to be considered or the BIND() function allowing the creation of a new variable based on the retrieved characteristic of a single result row (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013). Although the MINUS and UNION operators can be used as in SQL, the INTERSECT operator is useless and is forsaken in SPARQL and the MINUS and UNION operators should be preceded and followed by statements between curly brackets like the WHERE clause (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013). SPARQL has also the advantage to allow including entries where a set of statements in the WHERE clause is not respected by putting these statements after the OPTIONAL operator between curly brackets (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013).

In Wikidata, the Wikidata Query Service (https://www.wikidata.org) allows to query the knowledge graph using SPARQL (Malyshev, et al., 2018; Turki, et al., 2019). The required Wikidata prefixes are already supported in the backend of the service and do not need to be defined (Malyshev, et al., 2018). What the user needs to do is to formulate their SPARQL query (Black in Fig. 5) and click on the Run button (Blue in Fig. 5). After a compilation period, the results will appear (Green in Fig. 5) and can be downloaded in different formats (Brown in Fig. 5), including JSON, TSV, CSV, HTML, and SVG. Different modes for the visualization of the query results can be chosen (Purple in Fig. 5), particularly table, charts (line, scatter, area, bubble), image grid, map, tree, timeline, and graph. The query service also allows users to use a query helper (Red in Fig. 5) that can generate basic SPARQL queries and get inspired by sample queries (Yellow in Fig. 5), especially when they lack experience. It also allows us to generate a short link for the query (Pink in Fig. 5) and codes to embed the query results in web pages and computer programs (Brown in Fig. 5) (Malyshev, et al., 2018).

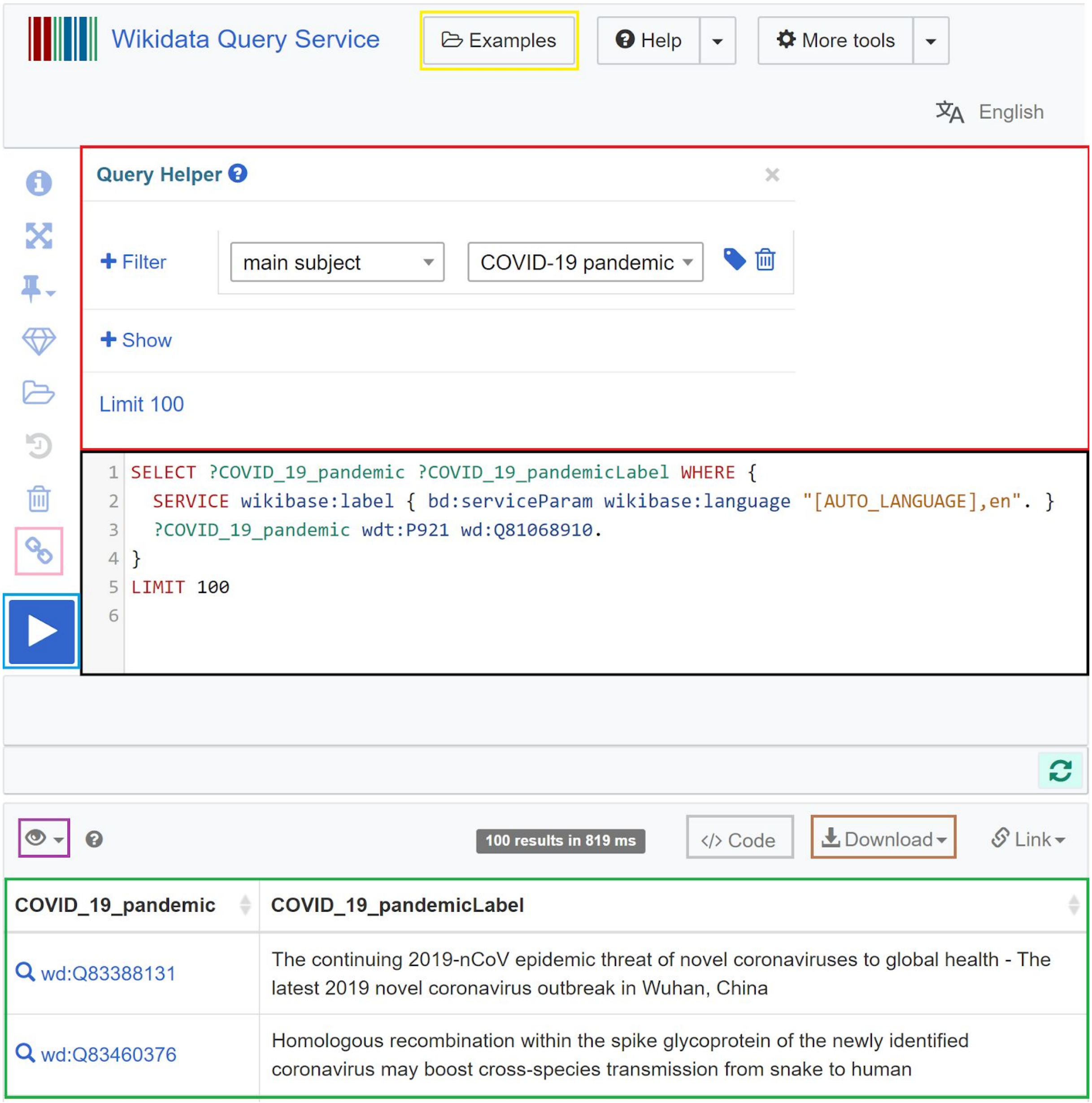

Figure 5: Web interface of Wikidata Query Service [Source: https://w.wiki/aeH, Derived from: https://query.wikidata.org, License: CC BY-SA 4.0]. It involves a query field (Black), a query builder (Red), a short link button (Pink), a Run button (Blue), a visualization mode button (Purple), a download button (Brown), an embedding code generation button (Grey), a results field (green), and a sample query button (Yellow)

The statements in the WHERE clause should be defined such that known subjects and objects are preceded by *wd* prefix whatever they are Wikidata items or properties and that the predicate should be a Wikidata property and it is preceded by *wdt* prefix as clearly shown in Fig. 6. Other Wikidata prefixes can be used to parse Wikidata qualifiers (*pq* and *pqv*) and references (*pr* and *prv*) or to link between a Wikidata statement to one of its components (*p*, *prov*, *ps*, and *psv*). The *wikibase* prefix can be used to return the characteristics of an item, a property or a statement. For example, *wikibase:directClaim* and *wikibase:Claim* can shift a property from a Wikidata prefix to another one (e.g. shifting Wikidata properties from *wdt* to *wd*), and *wikibase:rank* can be useful to return the level of importance assigned by the community to a statement.

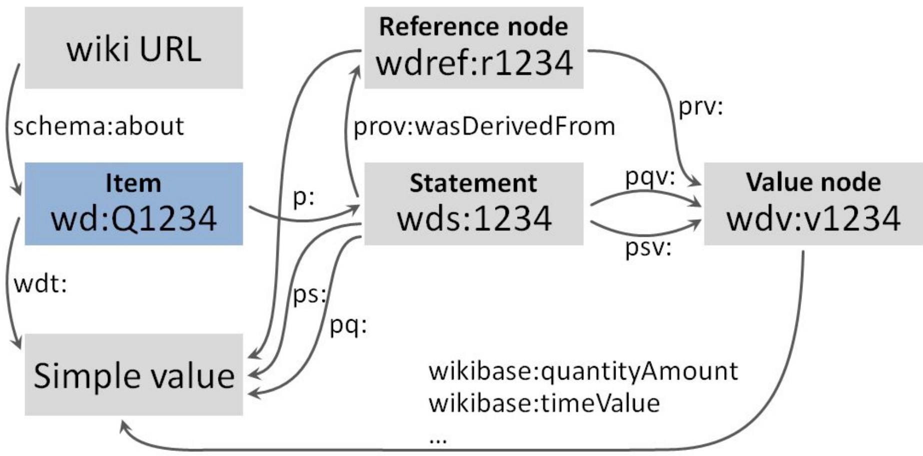

Figure 6: RDF data structure of Wikidata knowledge graph [Available at: https://w.wiki/any, adapted from source: https://w.wiki/ZUA, Michael F. Schönitzer, CC-BY 4.0]

5. Constraint-driven inference of biomedical property constraints

As described above, Wikidata properties are assigned property constraints and statements as logical conditions for the use of the types of triples to represent knowledge in Wikidata (Fig. 2). Screening Wikidata items in a class to identify common features of the assessed entities based on a set of formal rules has been previously proposed (Marx & Krötzsch, 2017; Hanika et al. 2019). These features involve common characteristics of the data model of the concerned class as well as patterns of used Wikidata properties such as symmetry and are later used to verify the completeness of the class and validate the statements related to the evaluated class using SPARQL queries. In this work, we propose a similar protocol fully based on logical constraints fully implementable using SPARQL queries to infer constraints for the assessment of the usage of relation types (*P*) on Wikidata based on the most frequently used corresponding inverse statements (C_O, P^{-1}, C_S). These constraints can be later used to define COVID-19-related Wikidata statements and to generate ShEx schemas for COVID-19-related Wikidata classes. Fig. 7 represents the scheme of the given relation type that will be used to assess and validate the use of Wikidata properties.

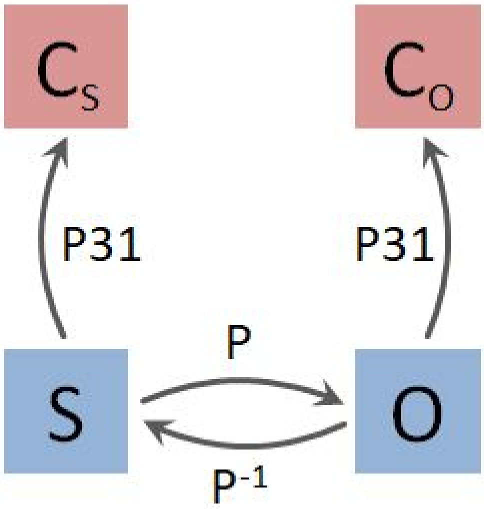

Figure 7: Scheme of a given Wikidata property [Source: https://w.wiki/anw, License: CC BY 4.0]: *S* and *O* are respectively the subject and the object of the statement, *P* is the predicate of the statement, P^{-1} is the inverse property of *P*, C_S is the class of the subject, and C_O is the class of the object.

Once retrieved, the common inverse property statements (C_O, P^{-1}, C_S) of the given Wikidata property *P* will be used to identify relations that use *P* in an uncommon and probably wrong way, to identify missing inverse relations of *P*(*S*,*O*) corresponding to the most used (C_O, P^{-1}, C_S) scheme, and to identify the Wikidata items missing statements using *P* as shown in Table 3. The assessment of the usage of the given Wikidata property will not be restricted to these tasks, as it also involves the identification of relations using *P* not supported by references and the identification of Wikidata properties used to define references for relations using *P*.

Table 3: Tasks for quality assessment of the usage of Wikidata relation types using the Wikidata SPARQL endpoint

Task	Description
Defining the scheme of a Wikidata property	
T1	Identify common use cases[19] of *P*: (C_S,C_O) pairs
T2	Identify inverse properties of *P* corresponding to each common use case: (C_S, R^{-1},C_O) statements
Identifying the deficiencies of the scheme	
T3	For each returned P^{-1}, identify *P*(*S*,*O*) relations supported by references and corresponding to the most common (C_S, P^{-1}, C_O) statement but not available in Wikidata
T4	Identify *P*(*S*,*O*) relations not corresponding to the most common scheme of *P*
Assessing the reference support of relations using the studied Wikidata property	
T5	Identify Wikidata properties used to define references for relations using *P*

This task set is useful to assess and adjust the reference support, the language support, the quantity, and the quality of the relations using *P* and P^{-1} at a given point in time and can be easily completed using the Wikidata Query Service. The SPARQL query of each task is given in Appendix A, where <PropertyID> is the Wikidata ID of the studied property *P*,

[19] Use case: A set of conditions for the use of a relation type *P*.

<SubjectID> is the Wikidata ID of the subject class C_S that is most used with this property, and <ObjectID> is the Wikidata ID of the most used object class C_O.

For Tasks T1 and T2, we eliminated property use cases where classes C_S and C_O are first-order metaclasses (Q24017414), so that we do not get nonspecific use cases. Additionally, we only considered use cases applied to more than a defined usage threshold (here set as 100 but can change according to context) in order to omit statements that are not widely used in Wikidata. For Task T4, we used logical constraints to find statements where the subject is not an instance of the most used subject class C_S (G1), then to find statements where the object is not an instance of the most used object class C_O (G2). After that, we identified the statements that exist in both G1 and G2 as the most likely wrong statements (G1 ∩ G2) as they correspond neither to the most used subject class nor to the most used object class of the studied property. For Task T5, Wikidata properties used to define fewer than a threshold number of references using *P* were not considered (again, here set to 100). Our analysis was performed on September 20, 2019, following the Zika outbreak as a proactive action to build the data model infrastructure to support clinical information about future infectious epidemics in Wikidata (the date is relevant due to the rapidly expanding nature of the database).

To assess the effectiveness of the use of logical constraints to generate condition for the verification and validation of the use of relation types to enrich the Wikidata ontology, we applied our method to the main six Wikidata properties that can be used to represent COVID-19-related knowledge: *drug used for treatment* [P2176], *route of administration* [P636], *therapeutic area* [P4044], *significant drug interaction* [P7696], *medical condition treated* [P2175], and *symptoms* [P780]. Further details about these properties can be found in Table 4.

Table 4: Wikidata properties assessed in this study

Property	Description	Statements
Drug used for treatment (P2176)	drug, procedure, or therapy that can be used to treat a medical condition	6344
Significant drug interaction (P769)	clinically significant interaction between two pharmacologically active substances (i.e., drugs and/or active metabolites) where concomitant intake can lead to altered effectiveness or adverse drug events.	1850
Medical condition treated (P2175)	disease that this pharmaceutical drug, procedure, or therapy is used to treat	6499
Symptoms (P780)	possible symptoms of a medical condition	8068
Route of administration (P636)	path by which a drug, fluid, poison, or other substance is taken into the body	2900
Therapeutic area (P4044)	disease area in which a medical intervention is applied	1320

Task T1 was effective at sorting the common use cases of the studied Wikidata properties as shown in Table 5. All the retrieved use cases were proven to be logically accurate when compared to the descriptions of Wikidata properties available in Table 4. The most common use cases for *drugs used for treatment* [P2176], *therapeutic area* [P4044], *significant drug interactions* [P769], or *medical condition treated* [P780] corresponded to 72 percent or more of the supported statements. However, there was a significant lack of availability of common use cases for *route of administration* [P636] and *symptoms* [P780], and this can be explained by a deficiency of logical reasoning by users when using these properties due to human faults, to inexperience with Wikidata or to inconsistencies between languages, which often derive from slight differences in the naming and framing of Wikipedia articles in a given set of languages. This can be also explained by a current deficiency in Wikidata taxonomy in attributing Wikidata items to corresponding classes.

Table 5: Common use cases of the studied Wikidata properties

Wikidata ID	Property	Subject Class	Object Class	Number of Statements
P2176	Drug used for treatment	Disease (Q12136)	medication (Q12140)	4777
		Disease (Q12136)	essential medicine (Q35456)	1558
		Infectious disease (Q18123741)	medication (Q12140)	558
		Disease (Q12136)	Heterocyclic compound (Q193430)	484
		Disease (Q12136)	Biopharmaceutical (Q679692)	471
P636	Route of administration	medication (Q12140)	route of administration (Q621636)	179
P4044	Therapeutic area	Pharmaceutical product (Q28885102)	disease (Q12136)	1147
		mixture (Q169336)	disease (Q12136)	1142
		Pharmaceutical product (Q28885102)	rare disease (Q929833)	142
		mixture (Q169336)	rare disease (Q929833)	141
		Pharmaceutical product (Q28885102)	Designated intractable/rare diseases (Q42303753)	115
P769	Significant drug interaction	medication (Q12140)	medication (Q12140)	1729
		medication (Q12140)	essential medicine (Q35456)	524
		essential medicine (Q35456)	medication (Q12140)	507
		medication (Q12140)	Heterocyclic compound (Q193430)	342
		Heterocyclic compound (Q193430)	medication (Q12140)	338
P2175	Medical condition treated	medication (Q12140)	Disease (Q12136)	4729
		essential medicine (Q35456)	Disease (Q12136)	1520
		medication (Q12140)	Infectious disease (Q18123741)	557
		Heterocyclic compound (Q193430)	Disease (Q12136)	487
		Biopharmaceutical (Q679692)	Disease (Q12136)	449
P780	Symptoms	disease (Q12136)	symptom (Q169872)	338
		disease (Q12136)	disease (Q12136)	264

Task T2 successfully sorted the inverse properties of Wikidata relation types for each corresponding use case as shown in Table 6. Here, we found that three relations had clear inverse properties: *medical condition treated* [P2175], *significant drug interaction* [P769] and *drug used for treatment* [P2176] are the inverse properties, respectively, for *drug used for treatment* [P2176], *significant drug interaction* [P769] and *medical condition treated* [P2175]. Hence, P2175 and P2176 are inverse to each other, and P769 is inverse to itself. However, we did not find any common inverse properties for *route of administration* [P636], *therapeutic area* [P4044] or *symptoms* [P780]. Consequently, the Task T2 can be used not only to find inverse properties of Wikidata relation types but also to identify Wikidata relation types where inverse properties do not exist or are not used as intended. In such a situation, the user should manually search for any inverse property to verify whether it exists or propose to the Wikidata community to create it as a new property[20] if it does not exist (Turki, et al., 2019).

Table 6: Inverse properties corresponding to each common use case of the studied Wikidata relation types

Wikidata ID	Property	Inverse property	Use case		Number of Statements
			Subject Class	Object Class	
P2176	Drug used for treatment	medical condition treated (P2175)	Disease (Q12136)	Medication (Q12140)	4576
		medical condition treated (P2175)	Disease (Q12136)	essential medicine (Q35456)	1482
		medical condition treated (P2175)	Infectious disease (Q18123741)	Medication (Q12140)	549
		medical condition treated (P2175)	Disease (Q12136)	Heterocyclic compound (Q193430)	477
		medical condition treated (P2175)	Disease (Q12136)	Biopharmaceutical (Q679692)	442
P636	Route of administration	NA			
P4044	Therapeutic area	NA			
P769	Significant drug interaction	Significant drug interaction (P769)	Medication (Q12140)	Medication (Q12140)	1330
		Significant drug interaction (P769)	Medication (Q12140)	essential medicine (Q35456)	359
		Significant drug interaction (P769)	essential medicine (Q35456)	Medication (Q12140)	359
		Significant drug interaction (P769)	Heterocyclic compound (Q193430)	Medication (Q12140)	288
		Significant drug interaction (P769)	Medication (Q12140)	Heterocyclic compound (Q193430)	288
P2175	Medical condition treated	Drug used for treatment (P2176)	Medication (Q12140)	Disease (Q12136)	4576
		Drug used for treatment (P2176)	essential medicine (Q35456)	Disease (Q12136)	1482
		Drug used for treatment (P2176)	Medication (Q12140)	Infectious disease (Q18123741)	549
		Drug used for treatment (P2176)	Heterocyclic compound (Q193430)	Disease (Q12136)	477
		Drug used for treatment (P2176)	Biopharmaceutical (Q679692)	Disease (Q12136)	442
P780	Symptoms	NA			

Task T3 effectively extracted those statements that use *drug used for treatment* [P2176], *significant drug interaction* [P769] and *medical condition treated* [P2175] as a Wikidata relation type where related inverse relations do not exist in Wikidata as clearly stated in Table 7. Only relations corresponding to the most common use case of the related

[20] e.g. Risk factor property proposal: https://www.wikidata.org/wiki/Wikidata:Property_proposal/risk_factor

Wikidata property and supported by references are considered. For the studied Wikidata relation types, 688 missing inverse statements were identified. These statements can be directly added to Wikidata using tools for the automatic enrichment of Wikidata, particularly QuickStatements (Turki, et al., 2019), as they are supported by external references and are already stated in a Wikidata-friendly format.

Table 7: Number of missing inverse statements of Wikidata relations supported by references and corresponding to the most used scheme of each Wikidata property

Wikidata ID	Property	Most used scheme			Missing inverse statements
		Inverse property	Subject Class	Object Class	
P2176	Drug used for treatment	medical condition treated (P2175)	Disease (Q12136)	Medication (Q11173)	160
P636	Route of administration	NA			
P4044	Therapeutic area	NA			
P769	Significant drug interaction	Significant drug interaction (P769)	Medication (Q12140)	Medication (Q12140)	385
P2175	Medical condition treated	Drug used for treatment (P2176)	Medication (Q12140)	Disease (Q12136)	143
P780	Symptoms	NA			

Task T4 efficiently identified the statements not corresponding to the most common use case of the related Wikidata property as shown in Table 8. In fact, 11236 statements not corresponding to the most used subject class of the studied Wikidata properties and 7354 statements not corresponding to the most used object class of the studied Wikidata properties were identified.

Table 8: Number of statements not corresponding to the most common use case of each Wikidata property: Statements where the subject class is not the most used one (G1), statements where the object class is not the most used one (G2)

Wikidata ID	Property	G1	G2	G1∩G2
P2176	Drug used for treatment	858	390	72
P636	Route of administration	2656	1255	1255
P4044	Therapeutic area	5	171	3
P769	Significant drug interaction	82	42	3
P2175	Medical condition treated	620	1036	135
P780	Symptoms	7015	4460	3749

Among these statements, 5217 relations corresponded neither to the most common subject class nor to the most common object class of the considered properties. These results may be wrong statements that should be adjusted or deleted. However, they can also result from the lack of completeness of Wikidata taxonomy (i.e. a significant lack in defining relations between Wikidata items and corresponding classes). When applying expert validation to 800 randomly selected relations among the 5217 studied ones, we found that only 6.6% of these relations (53) were truly inaccurate and that the remaining 93.4% (747) were accurate but identified due to the lack of assignment of their subjects and objects to their hypernyms. The precision rate of the identification of deficient relations using this method seems to vary considering the studied property but does not exceed 10% (Fig. 8).

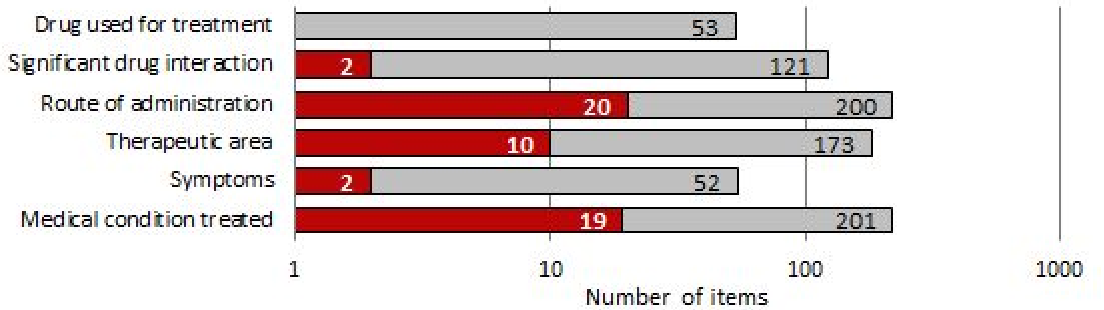

Figure 8: Relations returned by Task T4 for the studied Wikidata properties [Available at: https://w.wiki/ao2, License: CC BY 4.0]. Extracted relations verified by expert validation as deficient are represented in red. Note: log x-axis.

Accordingly, the results sorted by Task T4 should be manually verified and validated by experts, so that users can use true identified relations (False positive) to enrich their respective subject and object Wikidata items with corresponding missing classes and find the reasons behind the deficiency of wrong identified relations (True positive) to develop automatic methods to solve them. The insufficiencies of wrong relations can either be due to ontological reasons (64%) or medicine-related reasons (36%) as shown in Fig. 9 and cannot consequently be handled only by computer scientists. Efforts in crowdsourcing ontology verification of other biomedical ontologies such as SNOMED-CT confirmed the existence of both types of errors and stipulated that not adjusting these lexical resources and using them in clinical decision support can generate harmful recommendations (Mortensen, et al., 2014).

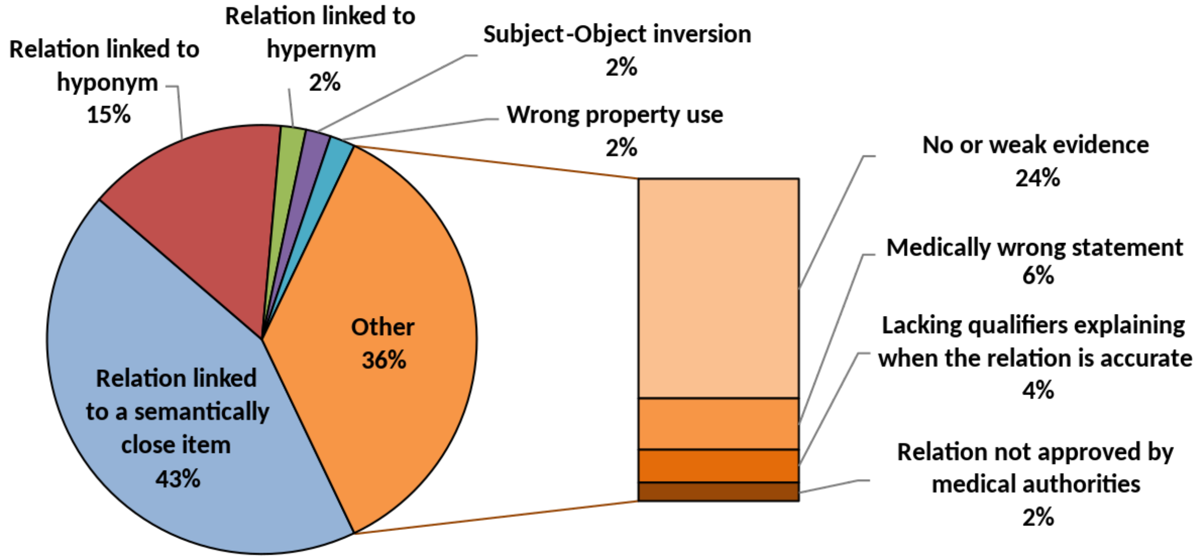

Figure 9: Reasons of the inaccuracy of the truly deficient identified relations (True positive) [Source: https://w.wiki/ao3, License: CC BY 4.0]

Task T5 was efficient in finding the Wikidata properties used to define the references of the statements for each studied relation type (Table 9). For the studied Wikidata relation types, we found that references are mainly defined using three properties: *stated in* [P248], *retrieved* [P813], and *reference URL* [P854]. One of the highest priority tasks on Wikidata is for experts to find and add appropriate references using these three properties to currently unsupported Wikidata relations. Once the references are in the system, further refinement is possible, e.g. a *reference URL* [P854] containing (or pointing to a page that contains) an external identifier for which Wikidata has a suitable property - e.g. *Digital Object Identifier* [P356] - then that property could be added to an item about the cited references, and the P854 statement replaced by a P248 statement pointing to that item.

Table 9: Wikidata properties used to define references for studied Wikidata relation types: *stated in* (P248), *retrieved* (P813), *language of work or name* (P407), *National Drug File Reference Terminology ID* (P2115), *reference URL* (P854), and *European Medicines Agency product number* (P3637)

Wikidata ID	Property	P248	P813	P407	P2115	P854	P3637
P2176	Drug used for treatment	6654	6636	3617	3522	1626	
P636	Route of administration					2647	
P4044	Therapeutic area	1310	1310			1313	1310
P769	Significant drug interaction	1757					
P2175	Medical condition treated	6683	6672	3533	3516	1719	
P780	Symptoms	257	114			7094	

6. Constraint-driven heuristics-based validation of epidemiological data

The characterization of epidemiological data is possible using a variety of statistical measures that show the acuteness, the dynamics, and the prognosis of a given disease outbreak. These measures include the simple cumulative count of cases (P1603 [199569 statements], noted *c*, as defined before), deaths (P1120 [243250 statements[21]], noted *d*), recoveries (P8010 [36119 statements], noted *r*), clinical tests (P8011 [21249 statements], noted *t*), and hospitalized cases (P8049 [5755 statements], noted *h*) as well as several measurements done by the synthesis of the values of simple epidemiological counts such as case fatality rate (P3457 [51504 statements], noted *m*), basic reproduction number (P3492, noted R_0), minimal incubation period in humans (P3488, noted *mn*), and maximal incubation period in humans (P3487, noted *mx*) (Rothman, Greenland, & Lash, 2008). For all these statistical data, every information should be coupled by a *point in time* (P585, noted *Z*) qualifier defining the date of the stated measurement and by a *Determination method* (P459, noted *Q*) qualifier identifying the measurement method of the given information as these variables are subject to change over days or according to used methods of computation.

Thanks to the logic behind simple count statistics (*c*, *t*, *d*, *h*, and *r* statements), several conditions based on the comparison of epidemiological variables of a regional COVID-19

[21] As of August 8, 2020. For updated statistics, https://w.wiki/Z5m.

disease outbreak for a given date *Z*, the comparison of the statistical variation of an epidemiological variable over days, and the comparison between the epidemiological values of a general disease outbreak with the ones of its components (each defined as a *part of* [P361] of the general outbreak) as shown in Table 10. Tasks V1 and V2 have been generated from the evidence that COVID-19 started in late 2019 and that its clinical discovery can only be done through medical diagnosis techniques (Zu, et al., 2020). Tasks V3 and V4 have been derived from the fact that *c*, *d*, *r,* and *t* are cumulative counts. Consequently, these variables are only subjects to remain constant or increase over days. Task V5 is motivated by the fact that a simple epidemiological count cannot return negative values. Tasks V6, V7, V8, and V9 are due to the evidence that *d*, *r*, and *h* cannot be superior to *c* as a patient needs to be affected by SARS-CoV-2 to die or be hospitalized due to the contraction of COVID-19 (Rothman, Greenland, & Lash, 2008) and that a patient needs to undergo COVID-19 testing to be confirmed as a case of the disease (Zu, et al., 2020). V10 is built upon the assumption that *c*, *d*, *r*, *h,* and *t* values can be geographically aggregated (Rothman, Greenland, & Lash, 2008).

Table 10: Tasks for the heuristics-based evaluation of epidemiological data using the Wikidata SPARQL endpoint

Task	Description
Validating qualifiers of COVID-19 epidemiological statements	
V1	Verify *Z* as a date > November 01, 2019
V2	Verify *Q* as any subclass of (P279*) of medical diagnosis (Q177719)
Ensuring the cumulative pattern of *c*, *d*, *r*, and *t*	
V3	Identify *c*, *d*, *r* and *t* statements having a value in date *Z*+1 not superior or equal to the one in date *Z* (Verify if $d_Z \leq d_{Z+1}$, $r_Z \leq r_{Z+1}$, $t_Z \leq t_{Z+1}$, and $c_Z \leq c_{Z+1}$)
V4	Find missing values of *c*, *d*, *r* and *t* in date *Z*+1 where corresponding values in dates *Z* and *Z*+2 are equal
Validating values of epidemiological data for a given date	
V5	Identifying *c*, *d*, *r*, *h*, and *t* statements with negative values
V6	Identify *h* statements having a value superior to the number of cases for a date *Z*
V7	Identify *c* statements having a value superior or equal to the number of clinical tests for a date *Z*
V8	Identify *c* statements having a value inferior to the number of deaths for a date *Z*
V9	Identify *c* statements having a value inferior to the number of recoveries for a date *Z*
V10	Comparing the epidemiological variables of a general outbreak with the ones of its components

This task set has easily been applied using ten simple SPARQL queries that can be found in Appendix B where <PropertyID> is the Wikidata property to be analyzed and has returned 5496 deficiencies in the COVID-19 epidemiological information as shown in Table 11. Among these mistaken statements, 2856 were *number of cases* statements, 2467 were *number of deaths* statements, 189 were *number of recoveries* statements, 9 were *number of clinical tests* statements, and 10 were *number of hospitalized cases* statements. This distribution of the deficiencies among epidemiological properties is explained by the dominance of *number of cases* and *number of deaths* statements on the COVID-19 epidemiological information. Most of these mistakes are linked to a violation of the cumulative pattern of major variables. These deficiencies can be removed using tools for the automatic enrichment of Wikidata like QuickStatements (cf. Turki, et al., 2019) or adjusted one by one by active members of WikiProject COVID-19.

Table 11: Number of deficient statements for every type of epidemiological Wikidata property identified by each task (As of August 8, 2020)

	c	*d*	*r*	*t*	*h*	**Overall**
V1	18	9	10	2	1	40
V2	2	91	6	0	0	99
V3	660	92	6	5		763
V4	2081	2247	149	1		4478
V5	0	0	0	0	0	0
V6	8				8	8
V7	1			1		1
V8	9	9				9
V9	17		17			17
V10	60	19	1	0	1	81
Overall	2856	2467	189	9	10	5496

Concerning the variables issued from the integration of basic epidemiological counts (*m*, R_0, *mn* and *mx* statements), the situation is more complicated due to the complexity of the definition of these variables (Delamater, et al., 2019; Backer, Klinkenberg, & Wallinga, 2020; Li, et al., 2020). The basic reproduction number (R_0) is meant to be a constant that characterizes the dissemination power of an infectious disease. It is defined as the expected number of people (within a community with no prior exposure to the disease) that can contract a disease via the same infected individual. This variable should exceed the threshold of 1 to define a contagious disease (Delamater, et al., 2019). Although R_0 can give an idea about the general behavior of an outbreak of a given disease, any calculated value depends on the model used for its computation (e.g. *SIR* Model) as well as the underlying data and is consequently a bit imprecise and variable from one study to another (Delamater, et al., 2019). That is why It is not reliable to use this variable to evaluate the accuracy of simple epidemiological counts for a given pandemic. The only heuristic that can be applied to this variable is to verify if its value exceeds 1 for diseases causing large outbreaks. The incubation period of a disease gives an overview of the silent time required by an infectious agent to become active in the host organism and cause notable symptoms (Backer, Klinkenberg, & Wallinga, 2020; Li, et al., 2020). This variable is very important as it reveals how many days an inactive case can spread the disease in the host's environment before the host is being symptomatically identified. As a result, it can give an idea about the contagiousness of the infectious disease and its basic reproduction number (R_0). However, the determination of the incubation period - especially for a novel pathogen - is challenging, as a patient often cannot identify with precision the day when they had been exposed to the disease, at least if they did not travel to an endemic region or had not been in contact with a person they knew to be infected. This factor was behind the measurement of falsely small incubation periods for COVID-19 at the beginning of COVID-19 epidemic in China (Backer, Klinkenberg, & Wallinga, 2020). Furthermore, the use of minimal (*mn*) and maximal (*mx*) incubation periods in Wikidata to epidemiologically describe a disease instead of the median incubation period is a source of a lack of accuracy of the extracted values (Backer, Klinkenberg, & Wallinga, 2020; Li, et al., 2020). In fact, minimal and maximal incubation periods for a given disease are obtained in the function of the mean ($\overline{X}$) and standard deviation (σ) of the measures of the confidence interval of observed incubation periods in patients. Effectively, *mn* is equal to $\overline{X} - z * \frac{\sigma}{\sqrt{n}}$ and *mx* is equal to $\overline{X} + z * \frac{\sigma}{\sqrt{n}}$ where *n* is the number of analyzed observations and *z* is a characteristic of the hypothetical statistical distribution and of the statistical confidence level adopted for the estimation (Altman, et al., 2013). As a consequence, *mn* and *mx* variables are modified according to the number of observations (*n*) with a smaller difference between the two variables for higher values of *n*. As well, the two measures also vary according to the used statistical distribution and that is why different values of *mn* and *mx* were reported for COVID-19 when applying different distributions (Weibull, gamma and log-normal distribution) using a confidence level of 0.95 on the same set of observed cases (Backer, Klinkenberg, & Wallinga, 2020). Similarly, the two variables can change according to the adopted confidence level (*p* - 1) when using the same statistical distribution where a higher confidence level is correlated with a higher difference between the calculated *mn* and *mx* values, as shown in Fig. 10 (Ward & Murray-Ward, 1999; Altman, et al., 2013). Given these reasons and despite the significant importance of the two measures, these two statistical variables cannot be used to evaluate statistical epidemiological counts for COVID-19 due to their lack of precision and difficulty of determination.

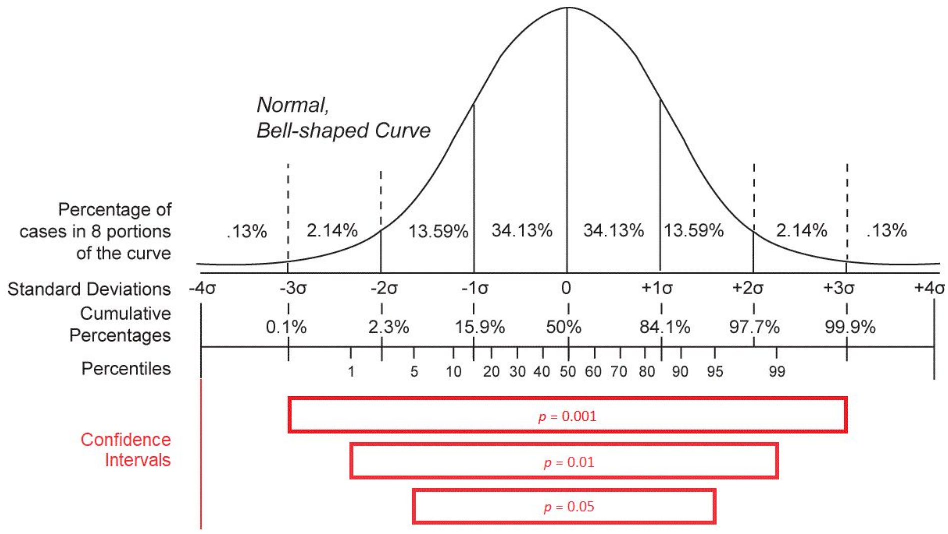

Figure 10: Confidence intervals for different p-values (*p*) when using a normal distribution [Source: https://w.wiki/aKT, License: Public Domain] (after Ward & Murray-Ward, 1999).

As for the reported case fatality rate (*m*), its definition is less intricate than the ones of the basic reproduction number and of the incubation period, as *m* is only the quotient of the cumulative number of deaths (*d*) by the cumulative number of cases (*c*) as stated in official reports. It is consequently easy to validate for a given disease by comparing its values with simple reported counts of cases and deaths (Rothman, Greenland, & Lash, 2008). Here, two simple heuristics can be applied using SPARQL queries as shown in Appendix C. As the number of deaths is less than or equal to the number of cases of a given disease, *m* values should be set between 0 and 1. That is why Task M1 is defined to extract *m* statements where $m > 1$ or $m < 0$. Also, as $m = d / c$ for a date *Z*, *m* values that are not close to the corresponding quotients of deaths by disease cases should be identified as deficient and *m* values should be stated for a given date *Z* if mortality and morbidity counts exist. Thus, Task M2 is created to extract m values where the absolute value of ($m - d/c$) is superior to 0.001, and Task M3 is developed to identify (item, date) pairs where *m* statements are missing and *c* and *d* statements are available in Wikidata. Absolute values for Task M2 are obtained using SPARQL's ABS function, and deficient (item, date) pairs are eliminated in Task M3 where $m > 1$ and $c < d$.

As a result of these three tasks, we interestingly identified 143 deficient *m* statements and 7116 missing *m* statements. 133 of the mistaken statements are identified thanks to Task M2 and concern 25 Wikidata items and 31 distinct dates and only 10 deficient statements related to 3 Wikidata items and 8 distinct dates are found using Task M1. These statements should be verified against reference datasets to verify their values and to determine the reason behind their deficiency. Such a reason can be the integration of the wrong case and death counts in Wikidata or a bug or inaccuracy within the source code of the bot making or updating such statements. The verification process can be automatically done using an algorithm that compares Wikidata values (*c*, *d* and *m* statements) with their corresponding ones in other databases (using file or API reading libraries) and subsequently adjusts statements using the Wikidata API directly or via tools like QuickStatements (Turki et al., 2019). As for the missing *m* statements returned by M3, they are linked to 395 disease outbreak items and to 205 distinct dates and concern 70% (7116/10168) of the (case count, death count) pairs available in Wikidata. The outcome of M3 proves the efficiency of comparative constraints to enrich and assess the completeness of epidemiological data available in a knowledge graph, particularly Wikidata, based on existing information. Consequently, derivatives of Task M3 can build to infer *d* values based on *c* and *m* statements or to find *c* values based on *d* and *m* statements. The missing statements found by such tasks can be integrated in Wikidata using a bot based on Wikidata API and Wikidata Query Service to ameliorate the completeness and integrity of available mortality data for epidemics, mainly the COVID-19 pandemic (Turki, et al., 2019).

7. Discussion

As shown, relational and statistical constraints have been demonstrated as efficient to identify use cases of a given relation type in a knowledge graph like Wikidata (Tables 5 and 6), to verify the completeness of inverse statements (Table 7), and to aid experts to find deficiencies within the taxonomy and the non-taxonomic relations of assessed knowledge graphs (Table 8 and Figures 8 and 9). This finding - combined with previous findings on the usefulness of SPARQL to find inconsistencies in semantic data based on known conditions particularly in the context of bioinformatics (Bolleman, et al., 2020; Marx & Krötzsch, 2017) - significantly proves the efficiency of rule-based approaches to evaluate semantic information from scratch by successfully addressing most of the competency questions, particularly conceptual orientation (*clarity*), coherence (*consistency*), strength (*precision*) and full coverage (*completeness*) with a similar accuracy as other available ontology evaluation algorithms (Amith, et al., 2019; Zhang & Bodenreider 2010). The scope of rule-based methods can be similarly expanded to cover other competency questions such as non-redundancy (*conciseness*) through the proposal of other logical constraints to tackle them such as a condition to find taxonomic relations to trim in a knowledge graph (Examples can be found at https://www.wikidata.org/wiki/Wikidata:Database_evaluation). The main limitation of applying the logical constraints using SPARQL in the context of Wikidata is that the runtime of a query that infers or verifies a complex condition or that analyzes a huge amount of class items or property use cases can exceed the timeout limit of the used endpoint (Malyshev, et al., 2018).

These evaluation assignments covered by our approach can be done by other rule-based (*structure-based* and *semantic-based*) ontology evaluation methods. Structure-based methods verify if a knowledge graph is defined according to a set of formatting constraints and semantic-based methods check if concepts and statements of a knowledge graph meet logical conditions (Amith, et al., 2018). Some of these methods are software tools, particularly Protégé extensions such as OWLET (Lampoltshammer & Heistracher, 2014) and OntoCheck (Schober, et al., 2012). OWLET infers the JSON schema logics of a given knowledge graph, converts them into OWL-DL axioms, and uses the semantic rules to validate the assessed ontological data (Lampoltshammer & Heistracher, 2014). OntoCheck screens an ontology to identify structural conventions and constraints for the definition of the analyzed relational information and consequently to homogenize the data structure and quality of the ontology by eliminating typos and pattern violations (Schober, et al., 2012). Here, the advantage of applying constraints using SPARQL is that its runtime is faster, as it does not require the download of the full dumps of the evaluated knowledge graph (Malyshev, et al., 2018). The benefit of our method and other structure-based and semantic-based web-based tools for knowledge graph validation like OntoKeeper (Amith, et al., 2019) and adviseEditor (Geller, et al., 2013) when compared to software tools is that the maximal size of the knowledge graphs that can be assessed by web services is larger than the one that can be evaluated by software tools because the latter depends on the requirements and capacities of the host computer (Lampoltshammer & Heistracher, 2014; Schober, et al., 2012). It is true that these drawbacks of other structure-based tools can be solved through the simplification of the knowledge graph by reducing redundancies using techniques like ontology trimming (Jantzen, et al., 2011) or through the construction of an abstraction network to decrease the complexity of the analyzed knowledge graph (Amith, et al., 2018; Halper, et al., 2015). However, knowledge graph simplification processes are time-consuming and resulting time gain can consequently be insignificant (Jantzen, et al., 2011; Amith, et al., 2018; Halper, et al., 2015).

Such tasks can be also solved using data-driven ontology evaluation methods. These techniques process texts in natural languages to validate the concepts and statements of a knowledge graph and currently include intrinsic (*lexical-based*) and extrinsic (*cross-validation, big data-based* and *corpus-based*) methods (Amith, et al., 2018). Lexical-based methods compare the terms and glosses of the items of a knowledge graph with the statements involving the analyzed items, mainly the taxonomic ones, to identify inconsistencies in the labels, descriptions or semantic relations of items (Amith, et al., 2018).

Lexical-based approaches use rules implemented in SQL or SPARQL to retrieve terms or glosses corresponding to a concept and their corresponding semantic relations, mostly *subclass of* statements (Rector & Iannone, 2012; Luo, Mejino Jr, & Zhang, 2013). The output is later analyzed using natural language processing techniques such as hamming distance measure (Luo, Mejino Jr, & Zhang, 2013), semantic annotation tools (Rector & Iannone, 2012) and semantic similarity measures (Amith, et al., 2018) to comparatively identify deficiencies in the semantic representation, labelling and symmetry of the assessed knowledge graph. Extrinsic data-based methods extract the usage and linguistic patterns from raw text corpuses such as bibliographic databases and clinical records (*Corpus-based methods*) or from gold standard semantic resources like large ontologies and knowledge graphs (*Cross-validation methods*) or from social media posts and interactions, Internet of Things data or web service statistics (*Big data-based methods*) (Amith, et al., 2018; Sebei, Hadj Taieb, & Ben Aouicha, 2018; Rector, Brandt, & Schneider, 2011; Gangemi, et al., 2005) using structure-based and semantic-based ontology evaluation methods as explained above (Rector, Brandt, & Schneider, 2011) as well as a range of techniques including machine learning (Bean, et al., 2017; Zhang, et al., 2018), topic modelling using latent dirichlet analysis (Abd-Alrazaq, et al., 2020), word embeddings (Zhang, et al., 2019), statistical correlations (Vanderkam, et al., 2013) and semantic annotation methods (Li, et al., 2016). The returned features of the analyzed resources are compared to the ones of the analyzed knowledge graph to assess the accuracy and completeness of the definition and use of concepts and properties (Amith, et al., 2018).

When compared to our proposed approach, lexical-based methods have the advantage to identify and adjust characteristics of a knowledge graph item based on its natural language information of a knowledge graph item, particularly terms and glosses (Rector & Iannone, 2012; Luo, Mejino Jr, & Zhang, 2013). The drawbacks of using semantic similarity, word embeddings and topic modelling approaches in such approaches is that these techniques are sensitive to the used parameters, to input characteristics and to the chosen models of computation and can consequently give different results according to the context of determination (Lastra-Díaz, et al., 2019; Hadj Taieb, Zesch, & Ben Aouicha, 2020). The current role of constraints in the extraction of lexical information and respective semantic relations (Rector & Iannone, 2012; Luo, Mejino Jr, & Zhang, 2013) proves that the scope of constraint-based validation should not only restricted to rule-based evaluation but also to lexical-based evaluation. Yet, the function of logical conditions should be expanded to refine the list of (lexical information, semantic relation) pairs to identify deficient and missing semantic relations and defective lexical data with a better accuracy and to support the processing of the multilingual lexical information in lexical-based methods as there are currently many SPARQL functions that are applied to analyze strings in knowledge graphs[22] such as STRLEN (length of a string), STRSTARTS (verification of a substring beginning a given string), STRENDS (verification of a substring finishing a given string), and CONTAINS (verification of a substring included in a given string) (DuCharme, 2013; Harris, Seaborne, & Prud'hommeaux, 2013).

[22] Detailed information about string functions in SPARQL can be found at https://www.w3.org/TR/sparql11-query/#func-strings .

As for the extrinsic data-driven methods, they are mainly based on large-scale resources that are regularly curated and enriched. Raw-text corpuses are mainly composed of scholarly publications (Raad & Cruz, 2015) and blog posts (Park, et al., 2016). Information in scholarly publications are ever changing according to the dynamic advances in scholarly knowledge, particularly medical data (Jalalifard, Norouzi, & Isfandyari-Moghaddam, 2013). This expansion of scientific information in scholarly publications is highly recognized in the context of COVID-19 where detailed information about COVID-19 disease and the SARS-CoV-2 virus is published within less than six months (Kagan, Moran-Gilad, & Fire, 2020). Big data is the set of real-time statistical and textual information that is generated by web services including search engines and social media and by Internet of Things objects including sensors (Sebei, Hadj Taieb, & Ben Aouicha, 2018). This data is characterized by its value, variety, variability, velocity, veracity and volume (Sebei, Hadj Taieb, & Ben Aouicha, 2018) and can be consequently used to track the changes of the community knowledge and consciousness over time (Abd-Alrazaq, et al., 2020; Turki, et al., 2020). Large semantic resources are ontologies and knowledge graphs that are built and curated by a community of specialists and that are regularly verified, updated and enriched using human efforts and computer programs (Lee, et al., 2013). These resources represent broad and reliable information about a given specialty through machine learning techniques (Zhang, et al., 2018) and the crowdsourcing of scientific efforts (Mortensen, et al., 2014) and can be consequently compared to other semantic databases for validation purposes. Examples of these resources are COVID-19 Disease Map (Ostaszewski, et al., 2020) and SNOMED-CT[23] (Li, et al., 2013). Given the dynamic characteristics of corpuses, big data and large-scale knowledge graphs, extrinsic data-driven methods can be more efficient than rule-based and lexical-based approaches, particularly the ones based on constraints, in identifying recent changes in the logical and semantic conditions for the definition of knowledge in a particular domain and accordingly adjusting the assessed knowledge graph (Amith, et al., 2018). Nonetheless, the growing and changing nature of gold standard resources require important human efforts (Mortensen, et al., 2014) and an advanced software architecture to maintain (e.g. structure-based and semantic-based methods), process (e.g. word embeddings and latent dirichlet analysis) and store (e.g. Hadoop and MapReduce) these reference resources on a daily basis (Li, et al., 2013; Sebei, Hadj Taieb, & Ben Aouicha, 2018). This architecture needs advanced hardware requirements and its results are subject to change according to used parameters (Sebei, Hadj Taieb, & Ben Aouicha, 2018). That is why constraint-based methods can be easier to apply than extrinsic data-driven knowledge graph evaluation methods.

These tasks are in line with the usage of Shape Expressions as well as property constraints and relations for the validation of data quality and completeness of the semantic information of class items in knowledge graphs as shown in the "Knowledge graph validation of Wikidata" section. A ShEx ShapeMap is a pair of a triple pattern for selecting entities to validate and a shape against which to validate them. This allows for the definition of the properties to be used for the items of a given class (Prud'hommeaux, Labra Gayo, & Solbrig, 2014; Waagmeester, et al., 2020a) and property constraints and relations based on the

[23] Systematized Nomenclature Of Medicine - Clinical Terms

meta-ontology (i.e. data skeleton) of Wikidata. Expressions written in shape-based property usage validation languages for RDF (e.g. SHACL) can be used to state conditions and formatting restrictions for the usage of relational and non-relational properties (Erxleben, et al., 2014; Thornton, et al., 2019; Gangemi, et al., 2005). SPARQL can be more efficient in inferring such information than the currently existing techniques that screen all the items and statements of a knowledge graph one by one to identify the conditions for the usage of properties (e.g. SQID) mainly because SPARQL is meant to directly extract information according to a pattern without having to evaluate all the conditions against all items of a knowledge graph (Marx & Krötzsch, 2017; Hanika, et al., 2019; Pérez, Arenas, & Gutierrez, 2009).

The separate execution of value-based constraints is common in the quality control of XML data. Typically, structural constraints are managed by RelaxNG or XML Schemas, while value-based constraints are captured as Schematron. Much as Schematron rules are typically embedded in RelaxNG, the consistency constraints presented above can be embedded in Shape Expressions Semantic Actions or in SHACL-SPARQL as shown in Fig. 11. These supplement structural schema languages with mechanisms to capture value-based constraints and in doing so, provide context for the enforcement of those constraints. The implementation of value-based constraints shown in the "Constraint-driven heuristics-based validation of epidemiological data" section can likewise be implemented in a shapes language (Labra-Gayo, et al., 2019). Parsing the rules in Table 3 and 10 would allow the mechanical generation or augmentation of shapes, providing flexibility for how the rules are expressed while still exploiting the power of shapes languages for validation.

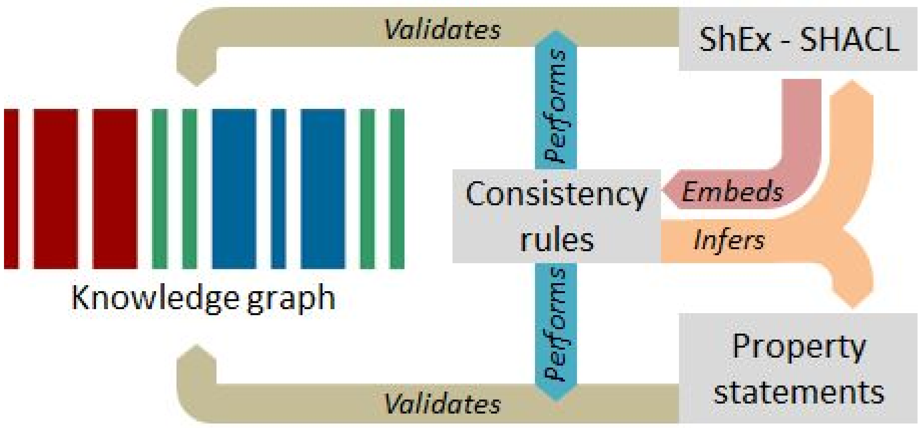

Figure 11: Interactions between consistency rules, property statements and RDF validation languages [Source: https://w.wiki/ao5, License: CC BY 4.0]

8. Conclusion

In this paper, we investigate how to best assess COVID-19 knowledge in collaborative ontologies and knowledge graphs (particularly Wikidata) using relational and statistical constraints. Collaborative databases produced through the cumulative edits of thousands of users are able to generate huge amounts of structured information (Turki, et al., 2019) but as a result of their entirely uncoordinated development, they often result in uneven coverage of crucial information and inconsistent expression of that information. The resulting gaps are a significant problem (false negatives, false positives, reasoning deficiencies, and missing references). Avoiding, identifying, and closing these gaps is therefore of top importance. We presented a standardized methodology for auditing key aspects of data quality and completeness for these resources[24].

This approach complements and informs shape-based methods for data conformance to community-decided schemas. The SPARQL execution does not require any pre-processing, and is not only restricted to the validation of the representation of a given item according to a reference data model but also to the comparison of the assessed relational and statistical statements. Our method is demonstrated as useful for measuring the overall accuracy and data quality on a subset of Wikidata and is consequently a necessary first step in any pipeline for detecting and fixing issues in collaborative ontologies and knowledge graphs. As a future direction, we will investigate how biomedical knowledge, particularly COVID-19 information are integrated to Wikidata and dynamically visualized using SPARQL queries, we will consider detailed information about the timely evolution of knowledge graphs in our methods for constraint-based validation by incorporating edit history SPARQL endpoint APIs of knowledge graphs (Pellissier Tanon & Suchanek, 2019, Dos Reis, Pruski, Da Silveira, & Reynaud-Delaître, 2014). We will also couple the information inferred using this method[25] with Shape Expressions and the explicit constraints of relation types to provide a more effective enrichment, refinement, and adjustment of collaborative ontologies and knowledge graphs.

Conflict of interest

All the co-authors of this paper except EP are active members of WikiProject Medicine, the community curating clinical knowledge in Wikidata, and of WikiProject COVID-19, the community developing multidisciplinary COVID-19 information in Wikidata. DJ is a non-paid voluntary member of the Board of Trustees of Wikimedia Foundation, the non-profit publisher of Wikipedia and Wikidata. EP is a co-creator of SPARQL. EP and JELG are co-creators of ShEx.

Acknowledgements

We thank the Wikidata community, Olivier Corby (Université Côte d'Azur, France), Odile Papini (Aix-Marseille Université, France), Egon Willighagen (Maastricht University, Netherlands), and Mahir Morshed (University of Illinois at Urbana-Champaign, United States of America) for useful comments and discussions about the topic of this research paper. The work done by Houcemeddine Turki, Mohamed Ali Hadj Taieb and Mohamed Ben Aouicha was supported by the Ministry of Higher Education and Scientific Research in Tunisia (MoHESR) in the framework of Federated Research Project PRFCOV19-D1-P1, by Wikimedia Foundation through a rapid grant, and by WikiCred Grants Initiative of Craig Newmark Philanthropies, Facebook, and Microsoft. The work done by Jose Emilio Labra Gayo was partially funded by the Spanish Ministry of Economy and Competitiveness (Society challenges: TIN2017-88877-R). The work done by Daniel Mietchen was supported in part by

[24] This method can be adapted to meet the needs of the user. For instance, the SPARQL queries can be slightly adjusted to assess other patterns in collaborative ontologies such as the usage of classes.

[25] This information can be represented in the form of RDF triples where the subject is the studied relation type and integrated to Wikidata.

the Alfred P. Sloan Foundation under grant number G-2019-11458. We also acknowledge that current policies at Tunisia's Ministry of Higher Education and Scientific Research stand in the way of publishing this manuscript as an open-access journal article.

References

Abd-Alrazaq, A., Alhuwail, D., Househ, M., Hamdi, M., & Shah, Z. (2020). Top concerns of tweeters during the COVID-19 pandemic: infoveillance study. *Journal of medical Internet research*, *22*(4), e19016. doi:10.2196/19016.

Addshore, Mietchen, D., & Willighagen, E. (2020). *Wikidata Queries around the SARS-CoV-2 virus and pandemic*. NL: Zenodo. doi:10.5281/zenodo.3977414.

Altman, D., Machin, D., Bryant, T., & Gardner, M. (Eds.). (2013). *Statistics with confidence: confidence intervals and statistical guidelines*. John Wiley & Sons. ISBN:978-0-727-91375-3.

Amith, M., He, Z., Bian, J., Lossio-Ventura, J. A., & Tao, C. (2018). Assessing the practice of biomedical ontology evaluation: Gaps and opportunities. *Journal of Biomedical Informatics, 80*, 1-13. doi:10.1016/j.jbi.2018.02.010.

Amith, M., Manion, F., Liang, C., Harris, M., Wang, D., He, Y., & Tao, C. (2019). Architecture and usability of OntoKeeper, an ontology evaluation tool. *BMC medical informatics and decision making*, *19*(4), 152. doi:10.1186/s12911-019-0859-z.

Backer, J. A., Klinkenberg, D., & Wallinga, J. (2020). Incubation period of 2019 novel coronavirus (2019-nCoV) infections among travellers from Wuhan, China, 20–28 January 2020. *Eurosurveillance*, *25*(5), 2000062. doi:10.2807/1560-7917.ES.2020.25.5.2000062.

Bansal, R., & Chawla, S. (2016). Design and development of semantic web-based system for computer science domain-specific information retrieval. *Perspectives in Science, 8*, 330–333. doi:10.1016/j.pisc.2016.04.067.

Bean, D. M., Wu, H., Iqbal, E., Dzahini, O., Ibrahim, Z. M., Broadbent, M., ... & Dobson, R. J. (2017). Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Scientific reports*, *7*(1), 1-11. doi:10.1038/s41598-017-16674-x.

Ben Aouicha, M., & Hadj Taieb, M. A. (2016). Computing semantic similarity between biomedical concepts using new information content approach. *Journal of biomedical informatics*, *59*, 258-275. doi:10.1016/j.jbi.2015.12.007.

Bolleman, J., de Castro, E., Baratin, D., Gehant, S., Cuche, B. A., Auchincloss, A. H., ... & Rivoire, C. (2020). HAMAP as SPARQL rules—A portable annotation pipeline for genomes and proteomes. *GigaScience*, *9*(2), giaa003. doi:10.1093/gigascience/giaa003.

Bonifati, A., Martens, W., & Timm, T. (2017). An analytical study of large SPARQL query logs. *Proceedings of the VLDB Endowment*, *11*(2), 149-161. doi:10.14778/3149193.3149196.

Brank, J., Grobelnik, M., & Mladenic, D. (2005). A survey of ontology evaluation techniques. *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)* (pp. 166–170). Ljubljana, Slovenia: Citeseer.

Burgstaller-Muehlbacher, S., Waagmeester, A., Mitraka, E., Turner, J., Putman, T., Leong, J., . . . Su, A. I. (2016). Wikidata as a semantic framework for the Gene Wiki initiative. *Database, 2016*, baw015. doi:10.1093/database/baw015.

Chen, Q., Lee, K., Yan, S., Kim, S., Wei, C. H., & Lu, Z. (2020). BioConceptVec: Creating and evaluating literature-based biomedical concept embeddings on a large scale. *PLoS computational biology*, *16*(4), e1007617. doi:10.1371/journal.pcbi.1007617.

Cuan-Baltazar, J. Y., Muñoz-Perez, M. J., Robledo-Vega, C., Pérez-Zepeda, M. F., & Soto-Vega, E. (2020). Misinformation of COVID-19 on the internet: infodemiology study. *JMIR public health and surveillance*, *6*(2), e18444. doi:10.2196/18444.

Degbelo, A. (2017). A Snapshot of Ontology Evaluation Criteria and Strategies. *Proceedings of the 13th International Conference on Semantic Systems* (pp. 1–8). New York: ACM. doi:10.1145/3132218.3132219.

Delamater, P. L., Street, E. J., Leslie, T. F., Yang, Y. T., & Jacobsen, K. H. (2019). Complexity of the basic reproduction number (R0). *Emerging infectious diseases*, *25*(1), 1. doi:10.3201/eid2501.171901.

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases*, *20*(5), 533-534. doi: 10.1016/S1473-3099(20)30120-1.

Dos Reis, J. C., Pruski, C., Da Silveira, M., & Reynaud-Delaître, C. (2014). Understanding semantic mapping evolution by observing changes in biomedical ontologies. *Journal of biomedical informatics, 47*, 71-82. doi:10.1016/j.jbi.2013.09.006

DuCharme, B. (2013). *Learning SPARQL: querying and updating with SPARQL 1.1*. O'Reilly Media, Inc. ISBN:978-1449306595.

Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., & Vrandečić, D. (2014). Introducing Wikidata to the Linked Data Web. *The Semantic Web – ISWC 2014* (pp. 50–65). Springer International Publishing. doi:10.1007/978-3-319-11964-9_4.

Färber, M., Bartscherer, F., Menne, C., & Rettinger, A. (2018). Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web, 9*(1), 77–129. doi:10.3233/SW-170275.

Farda-Sarbas, M., Zhu, H., Nest, M. F., & Müller-Birn, C. (2019). Approving automation: analyzing requests for permissions of bots in wikidata. In *Proceedings of the 15th International Symposium on Open Collaboration* (pp. 1-10). doi:10.1145/3306446.3340833.

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2005). A theoretical framework for ontology evaluation and validation. In *SWAP* (Vol. 166, p. 16).

Gyrard, A., Datta, S. K., & Bonnet, C. (2018). A survey and analysis of ontology-based software tools for semantic interoperability in IoT and WoT landscapes. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, (pp. 86–91). doi:10.1109/WF-IoT.2018.8355091.

Hadj Taieb, M. A., Zesch, T., & Ben Aouicha, M. (2020). A survey of semantic relatedness evaluation datasets and procedures. *Artificial Intelligence Review*, *53*(6), 4407-4448. doi:10.1007/s10462-019-09796-3.

Halper, M., Gu, H., Perl, Y., & Ochs, C. (2015). Abstraction networks for terminologies: supporting management of “big knowledge”. *Artificial intelligence in medicine*, *64*(1), 1-16. doi:10.1016/j.artmed.2015.03.005

Hanika, T., Marx, M., & Stumme, G. (2019). Discovering implicational knowledge in Wikidata. In *International Conference on Formal Concept Analysis* (pp. 315-323). Springer, Cham. doi:10.1007/978-3-030-21462-3_21.

Harris, S., Seaborne, A., & Prud’hommeaux, E. (2013). SPARQL 1.1 query language. *W3C recommendation*, *21*(10), 778.

Heftberger, A., Höper, J., Müller-Birn, C., & Walkowski, N.-O. (2020). Opening up Research Data in Film Studies by Using the Structured Knowledge Base Wikidata. In H. Kremers, *Digital Cultural Heritage* (pp. 401–410). Springer International Publishing. doi:10.1007/978-3-030-15200-0_27.

Heymann, D. L. (2020). Data sharing and outbreaks: best practice exemplified. *The Lancet, 395* (10223), 469-470. doi: 10.1016/S0140-6736(20)30184-7.

Hsu, P. Y., & Parker, D. S. (1995). Improving SQL with generalized quantifiers. In *Proceedings of the Eleventh International Conference on Data Engineering* (pp. 298-305). IEEE. doi:10.1109/ICDE.1995.380381.

Jalalifard, M., Norouzi, Y., & Isfandyari-Moghaddam, A. (2013). Analyzing web citations availability and half-life in medical journals. *Aslib Proceedings*, *65*(3), 242. doi:10.1108/00012531311330638.

Jantzen, S. G., Sutherland, B. J., Minkley, D. R., & Koop, B. F. (2011). GO Trimming: Systematically reducing redundancy in large Gene Ontology datasets. *BMC research notes*, *4*(1), 267. doi:10.1186/1756-0500-4-267.

Jemielniak, D. (2014). *Common knowledge?: An ethnography of Wikipedia.* Stanford: Stanford University Press. ISBN:978-0804789448

Jemielniak, D., & Wilamowski, M. (2017). Cultural diversity of quality of information on Wikipedias. *Journal of the Association for Information Science and Technology, 68*(10), 2460-2470. doi:10.1002/asi.23901.

Jemielniak, D., & Przegalinska, A. (2020) *Collaborative Society*, Cambridge, MA: MIT Press. ISBN:978-0262537919.

Kaffee, L. A., Piscopo, A., Vougiouklis, P., Simperl, E., Carr, L., & Pintscher, L. (2017). A glimpse into babel: An analysis of multilinguality in wikidata. *Proceedings of the 13th International Symposium on Open Collaboration* (p. 14). ACM. doi:10.1145/3125433.3125465.

Kaffee, L.-A., & Simperl, E. (2018). Analysis of Editors' Languages in Wikidata. *Proceedings of the 14th International Symposium on Open Collaboration* (p. 21). ACM. doi:10.1145/3233391.3233965

Kagan, D., Moran-Gilad, J., & Fire, M. (2020). Scientometric trends for coronaviruses and other emerging viral infections. *GigaScience, 9*(8), giaa085. doi:10.1093/gigascience/giaa085.

Knublauch, H., & Kontokostas, D. (2017, 6). *Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017.* W3C Recommendation, #w3c#. Retrieved from https://www.w3.org/TR/2017/REC-shacl-20170720/

Krishnan, L., Ogunwole, S. M., & Cooper, L. A. (2020). Historical Insights on Coronavirus Disease 2019 (COVID-19), the 1918 Influenza Pandemic, and Racial Disparities: Illuminating a Path Forward. *Annals of Internal Medicine*. doi:10.7326/M20-2223.

Kumar, A. P., Kumar, A., & Kumar, V. N. (2011). A comprehensive comparative study of SPARQL and SQL. *International Journal of Computer Science and Information Technologies*, *2*(4), 1706-1710.

Labra Gayo, J. E., & Alvarez Rodríguez, J. M. (2013). Validating statistical index data represented in RDF using SPARQL queries. *RDF Validation Workshop. Practical Assurances for Quality RDF Data.* Cambridge: http://www.w3.org/2012/12/rdf-val.

Labra Gayo, J. E., Prud'Hommeaux, E., Boneva, I., & Kontokostas, D. (2017). Validating RDF data. *Synthesis Lectures on Semantic Web: Theory and Technology*, *7*(1), 1-328. doi:10.2200/s00786ed1v01y201707wbe016.

Labra-Gayo, J. E., García-González, H., Fernández-Alvarez, D., & Prud'hommeaux, E. (2019). Challenges in RDF validation. In Current Trends in Semantic Web Technologies: Theory and Practice (pp. 121-151). Springer, Cham. doi:10.1007/978-3-030-06149-4_6.

Lastra-Díaz, J. J., Goikoetxea, J., Hadj Taieb, M. A., García-Serrano, A., Ben Aouicha, M., & Agirre, E. (2019). A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, *85*, 645-665. doi:10.1016/j.engappai.2019.07.010.

Lampoltshammer, T. J., & Heistracher, T. (2014). Ontology evaluation with Protégé using OWLET. *Infocommunications Journal*, *6*(2), 12-17.

Lanamäki, A., & Lindman, J. (2018). Latent Groups in Online Communities: a Longitudinal Study in Wikipedia. *Computer Supported Cooperative Work (CSCW)*, *27*(1), 77-106. doi:10.1007/s10606-017-9295-8.

Lee, D., Cornet, R., Lau, F., & De Keizer, N. (2013). A survey of SNOMED CT implementations. *Journal of biomedical informatics*, *46*(1), 87-96.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C. H., Leaman, R., ... & Lu, Z. (2016). BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, *2016*. doi:10.1093/database/baw068.

Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., ... & Xing, X. (2020). Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. *New England Journal of Medicine, 382*, 1199-1207. doi:10.1056/NEJMoa2001316.

Lozano-Tello, A., & Gomez-Perez, A. (2004). Ontometric: A Method to Choose the Appropriate Ontology. *Journal of Database Management (JDM), 15*(2), 1-18.

Luggen, M., Difallah, D., Sarasua, C., Demartini, G., & Cudré-Mauroux, P. (2019). Non-parametric Class Completeness Estimators for Collaborative Knowledge Graphs—The Case of Wikidata. *The Semantic Web – ISWC 2019* (pp. 453–469). Springer International Publishing. doi:10.1007/978-3-030-30793-6_26.

Luo, L., Mejino Jr, J. L., & Zhang, G. Q. (2013). An analysis of FMA using structural self-bisimilarity. *Journal of biomedical informatics*, *46*(3), 497-505. doi:10.1016/j.jbi.2013.03.005.

Malyshev, S., Krötzsch, M., González, L., Gonsior, J., & Bielefeldt, A. (2018). Getting the most out of wikidata: Semantic technology usage in wikipedia's knowledge graph. *International Semantic Web Conference* (pp. 376-394). Springer, Cham. doi:10.1007/978-3-030-00668-6_23.

Martin, P. A. (2018). Evaluating Ontology Completeness via SPARQL and Relations-between-Classes Based Constraints. *11th International Conference on the Quality of Information and Communications Technology (QUATIC)*, (pp. 255–263). doi:10.1109/QUATIC.2018.00045.

Marx, M., & Krötzsch, M. (2017). SQID: Towards Ontological Reasoning for Wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations Track*. CEUR Workshop Proceedings.

Mayr, P., Scharnhorst, A., Larsen, B., Schaer, P., & Mutschke, P. (2014). Bibliometric-enhanced information retrieval. *European Conference on Information Retrieval* (pp. 798-801). Springer, Cham. doi:10.1007/978-3-319-06028-6_99.

Mietchen, D., Hagedorn, G., Willighagen, E., Rico, M., Gómez-Pérez, A., Aibar, E., Rafes, K,, Germain, C., Dunning, A., Pintscher, L., & Kinzler, D. (2015). Enabling open science: Wikidata for research (Wiki4R). *Research Ideas and Outcomes*, *1*, e7573. doi: 10.3897/rio.1.e7573.

Mietchen, D., & Li, J. (2020). Quantifying the Impact of Data Sharing on Outbreak Dynamics (QIDSOD). *Research Ideas and Outcomes*, *6*, e54770. doi: 10.3897/rio.6.e54770.

Miquel-Ribé, M., & Laniado, D. (2018). Wikipedia Culture Gap: Quantifying Content Imbalances Across 40 Language Editions. *Frontiers in Physics, 6*, 54. doi:10.3389/fphy.2018.00054.

Mitraka, E., Waagmeester, A., Burgstaller-Muehlbacher, S., Schriml, L. M., Su, A. I., & Good, B. M. (2015). Wikidata: A platform for data integration and dissemination for the life sciences and beyond. *bioRxiv*, 031971. doi:10.1101/031971.

Mora-Cantallops, M., Sánchez-Alonso, S., & García-Barriocanal, E. (2019). A systematic literature review on Wikidata. *Data Technologies and Applications, 53*, 250–268. doi:10.1108/DTA-12-2018-0110.

Mortensen, J. M., Minty, E. P., Januszyk, M., Sweeney, T. E., Rector, A. L., Noy, N. F., & Musen, M. A. (2014). Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT. *Journal of the American Medical Informatics Association, 22*, 640-648. doi:10.1136/amiajnl-2014-002901.

Müller-Birn, C., Karran, B., Lehmann, J., & Luczak-Rösch, M. (2015). Peer-production System or Collaborative Ontology Engineering Effort: What is Wikidata? *Proceedings of the 11th International Symposium on Open Collaboration* (pp. 20:1–20:10). New York: ACM. doi:10.1145/2788993.2789836.

Nielsen, F. Å., Mietchen, D., & Willighagen, E. (2017). Scholia, scientometrics and wikidata. In *European Semantic Web Conference* (pp. 237-259). Springer, Cham. doi:10.1007/978-3-319-70407-4_36.

Nielsen, F. Å., Thornton, K., & Labra-Gayo, J. E. (2019). Validating Danish Wikidata lexemes. In *15th International Conference on Semantic Systems, SEMPDS 2019.* Karlsruhe: CEUR-WS.

Nielsen, F. Å., Mietchen, D., & Willighagen, E. (2017). Scholia, scientometrics and wikidata. In *European Semantic Web Conference* (pp. 237-259). Springer, Cham. doi:10.1007/978-3-319-70407-4_36.

Obrst, L., Ceusters, W., Mani, I., Ray, S., & Smith, B. (2007). The Evaluation of Ontologies. *Semantic Web*, 139–158. doi:10.1007/978-0-387-48438-9_8.

Ostaszewski, M., Mazein, A., Gillespie, M. E., Kuperstein, I., Niarakis, A., Hermjakob, H., ... & Schreiber, F. (2020). COVID-19 Disease Map, building a computational repository of SARS-CoV-2 virus-host interaction mechanisms. *Scientific data*, *7*(1), 1-4. doi:10.1038/s41597-020-0477-8.

Park, M. S., He, Z., Chen, Z., Oh, S., & Bian, J. (2016). Consumers' use of UMLS concepts on social media: diabetes-related textual data analysis in blog and social Q&A sites. *JMIR medical informatics*, *4*(4), e41.

Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web, 8*(3), 489-508. doi:10.3233/SW-160218.

Pellissier Tanon, T., & Suchanek, F. (2019). Querying the Edit History of Wikidata. *The Semantic Web: ESWC 2019 Satellite Events* (pp. 161–166). Springer International Publishing. doi:978-3-030-32327-1_32.

Pellissier Tanon, T., Bourgaux, C., & Suchanek, F. (2019). Learning how to correct a knowledge base from the edit history. In *The World Wide Web Conference* (pp. 1465-1475). doi:10.1145/3308558.3313584.

Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS), 34*(3), 16. doi:10.1145/1567274.1567278.

Piad-Morffis, A., Gutiérrez, Y., & Muñoz, R. (2019). A corpus to support ehealth knowledge discovery technologies. *Journal of biomedical informatics*, *94*, 103172. doi:10.1016/j.jbi.2019.103172.

Pillai, S., Soon, L.-K., & Haw, S.-C. (2019). Comparing DBpedia, Wikidata, and YAGO for Web Information Retrieval. *Intelligent and Interactive Computing* (pp. 525–535). Springer Singapore. doi:10.1007/978-981-13-6031-2_40.

Piscopo, A., & Simperl, E. (2018). Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proceedings of the ACM on Human-Computer Interaction, 2*(CSCW), 141:1–141:18. doi:10.1145/3274410.

Prud'hommeaux, E., Labra Gayo, J. E., & Solbrig, H. (2014). Shape Expressions: An RDF Validation and Transformation Language.In *Proceedings of the 10th International Conference on Semantic Systems* (pp. 32-40). doi:10.1145/2660517.2660523

Raad, J., & Cruz, C. (2015). A survey on ontology evaluation methods. *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (pp. 179-186). ACM. doi:10.5220/0005591001790186.

RDA COVID-19 Working Group (2020). *RDA COVID-19; recommendations and guidelines, 5th release 28 May 2020*. Research Data Alliance. doi:10.15497/RDA00046.

Rector, A. L., Brandt, S., & Schneider, T. (2011). Getting the foot out of the pelvis: modeling problems affecting use of SNOMED CT hierarchies in practical applications. *Journal of the American Medical Informatics Association*, *18*(4), 432-440. doi:10.1136/amiajnl-2010-000045.

Rector, A., & Iannone, L. (2012). Lexically suggest, logically define: Quality assurance of the use of qualifiers and expected results of post-coordination in SNOMED CT. *Journal of biomedical informatics*, *45*(2), 199-209. doi:10.1016/j.jbi.2011.10.002.

Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern epidemiology*. Lippincott Williams & Wilkins. ISBN:978-1451190052.

Samuel, J. (2017). Collaborative Approach to Developing a Multilingual Ontology: A Case Study of Wikidata. *Research Conference on Metadata and Semantics Research* (pp. 167–172). Springer. doi:10.1007/978-3-319-70863-8_16.

Sarasua, C., Checco, A., Demartini, G., Difallah, D., Feldman, M., & Pintscher, L. (2019). The evolution of power and standard Wikidata editors: comparing editing behavior over time to predict lifespan and volume of edits. *Computer Supported Cooperative Work (CSCW)*, *28*(5), 843-882. doi:10.1007/s10606-018-9344-y.

Schober, D., Tudose, I., Svatek, V., & Boeker, M. (2012). OntoCheck: verifying ontology naming conventions and metadata completeness in Protégé 4. *Journal of Biomedical Semantics*, *3*(Suppl 2), S4. doi:10.1186/2041-1480-3-S2-S4.

Sebei, H., Taieb, M. A. H., & Aouicha, M. B. (2018). Review of social media analytics process and big data pipeline. *Social Network Analysis and Mining*, *8*(1), 30. doi:10.1007/s13278-018-0507-0.

Thornton, K., Solbrig, H., Stupp, G. S., Labra Gayo, J. E., Mietchen, D., Prud'Hommeaux, E., & Waagmeester, A. (2019). Using Shape Expressions (ShEx) to share RDF data models and to guide curation with rigorous validation. *European Semantic Web Conference* (pp. 606-620). Springer. doi:10.1007/978-3-030-21348-0_39.

Turki, H. (2018). Citation analysis is also useful to assess the eligibility of biomedical research works for inclusion in living systematic reviews. *Journal of clinical epidemiology, 97*, 124-125. doi:10.1016/j.jclinepi.2017.11.002.

Turki, H., Hadj Taieb, M. A., & Ben Aouicha, M. (2018). MeSH qualifiers, publication types and relation occurrence frequency are also useful for a better sentence-level extraction of biomedical relations. *Journal of biomedical informatics, 83*, 217-218. doi:10.1016/j.jbi.2018.05.011.

Turki, H., Shafee, T., Hadj Taieb, M. A., Ben Aouicha, M., Vrandečić, D., Das, D., & Hamdi, H. (2019). Wikidata: A large-scale collaborative ontological medical database. *Journal of Biomedical Informatics, 99*, 103292. doi:10.1016/j.jbi.2019.103292.

Turki, H., Vrandečić, D., Hamdi, H., & Adel, I. (2017). Using WikiData as a Multi-lingual Multi-dialectal Dictionary for Arabic Dialects. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)* (pp. 437–442). IEEE. doi:10.1109/AICCSA.2017.115.

Turki, H., Hadj Taieb, M. A., Ben Aouicha, M., & Abraham, A. (2020). Nature or Science: what Google Trends says. *Scientometrics, 124*(2), 1367-1385. doi:10.1007/s11192-020-03511-8.

Vanderkam, D., Schonberger, R., Rowley, H., & Kumar, S. (2013). *Nearest neighbor search in google correlate*. Google Inc.

Vasanthapriyan, S., Tian, J., & Xiang, J. (2017). An Ontology-Based Knowledge Framework for Software Testing. *Knowledge and Systems Sciences* (pp. 212–226). Springer Singapore. doi:10.1007/978-981-10-6989-5_18.

Vrandečić, D. (2009). Ontology Evaluation. In R. S. S. Staab, *Handbook on Ontologies* (pp. 293–313). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-92673-3_13.

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78-85. doi:10.1145/2629489.

Vrandečić, D. (2020). Architecture for a multilingual Wikipedia. *arXiv preprint arXiv:2004.04733*.

Waagmeester, A., Schriml, L., & Su, A. I. (2019). Wikidata as a linked-data hub for Biodiversity data. *Biodiversity Information Science and Standards, 3*, e35206. doi:10.3897/biss.3.35206.

Waagmeester, A., Willighagen, E. L., Su, A. I., Kutmon, M., Labra Gayo, J. E., Fernández-Álvarez, D., et al. (2020a). A protocol for adding knowledge to Wikidata, a case report. *BioRxiv*. doi:10.1101/2020.04.05.026336.

Waagmeester, A., Stupp, G., Burgstaller-Muehlbacher, S., Good, B. M., Malachi, G., Griffith, O. L., ... & Keating, S. M. (2020b). Wikidata as a knowledge graph for the life sciences. *eLife*, *9*, e52614. doi:10.7554/eLife.52614.

Ward, A., & Murray-Ward, M. (1999). *Assessment in the classroom*. Wadsworth Publishing Company. ISBN:978-0534527044.

Walisadeera, A. I., Ginige, A., & Wikramanayake, G. N. (2016). Ontology Evaluation Approaches: A Case Study from Agriculture Domain. *Computational Science and Its Applications -- ICCSA 2016* (pp. 318–333). Springer International Publishing. doi:10.1007/978-3-319-42089-9_23.

Wasi, S., Sachan, M., & Darbari, M. (2020). Document Classification Using Wikidata Properties. *Information and Communication Technology for Sustainable Development* (pp. 729–737). Singapore: Springer. doi:10.1007/978-981-13-7166-0_73.

Wiśniewski, D., Potoniec, J., Ławrynowicz, A. & Keet, C. M. (2019). Analysis of Ontology Competency Questions and their formalizations in SPARQL-OWL. *Journal of Web Semantics, 59*, 100534. doi:10.1016/j.websem.2019.100534.

Xu, B., Kraemer, M. U., & Data Curation Group (2020). Open access epidemiological data from the COVID-19 outbreak. *The Lancet Infectious Diseases*. doi:10.1016/S1473-3099(20)30119-5.

Zangerle, E., Gassler, W., Pichl, M., Steinhauser, S., & Specht, G. (2016). An Empirical Evaluation of Property Recommender Systems for Wikidata and Collaborative Knowledge Bases. *Proceedings of the 12th International Symposium on Open Collaboration* (pp. 18:1–18:8). New York: ACM. doi:10.1145/2957792.2957804.

Zhang, G. Q., & Bodenreider, O. (2010). Large-scale, exhaustive lattice-based structural auditing of SNOMED CT. In *AMIA Annual Symposium Proceedings* (Vol. 2010, p. 922). American Medical Informatics Association.

Zhang, Y., Lin, H., Yang, Z., Wang, J., Zhang, S., Sun, Y., & Yang, L. (2018). A hybrid model based on neural networks for biomedical relation extraction. *Journal of biomedical informatics, 81*, 83-92. doi:10.1016/j.jbi.2018.03.011

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, *6*(1), 52:1-52:9. doi:10.1038/s41597-019-0055-0.

Zu, Z. Y., Jiang, M. D., Xu, P. P., Chen, W., Ni, Q. Q., Lu, G. M., & Zhang, L. J. (2020). Coronavirus disease 2019 (COVID-19): a perspective from China. *Radiology*, 200490. doi:10.1148/radiol.2020200490.

Appendix A: SPARQL queries for the inference of the usage constraints of relation types in Wikidata

<table>
<tr><th>Task</th><th>SPARQL query</th></tr>
<tr><td>T1</td><td><pre>SELECT ?Cs ?CsLabel ?Co ?CoLabel (COUNT(*) AS ?count)
WHERE {
 ?S wdt:<PropertyID> ?O.
 ?S wdt:P31 ?Cs.
 ?O wdt:P31 ?Co.
 FILTER NOT EXISTS {
   {?Cs wdt:P31 wd:Q24017414} UNION {?Co wdt:P31 wd:Q24017414}.
 }
 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?Cs ?CsLabel ?Co ?CoLabel
ORDER BY DESC(?count)
LIMIT 5</pre></td></tr>
<tr><td>T2</td><td><pre>SELECT ?Cs ?CsLabel ?P1 ?Co ?CoLabel (COUNT(*) AS ?count)
WHERE {
 ?S wdt:<PropertyID> ?O.
 ?O ?P1 ?S.
 ?S wdt:P31 ?Cs.
 ?O wdt:P31 ?Co.
 FILTER NOT EXISTS {
   {?Cs wdt:P31 wd:Q24017414} UNION {?Co wdt:P31 wd:Q24017414}.
 }
 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?Cs ?CsLabel ?P1 ?Co ?CoLabel
ORDER BY DESC(?count)
LIMIT 5</pre></td></tr>
<tr><td>T3</td><td><pre>SELECT ?S ?SLabel ?O ?OLabel ?statement ?p ?ref ?refLabel WHERE {
 ?S p:<PropertyID> ?statement.
 ?statement ps:<PropertyID> ?O
 ?S wdt:<PropertyID> ?O.
 ?S wdt:P31 wd:<SubjectID>.
 ?O wdt:P31 wd:<ObjectID>.
 ?statement prov:wasDerivedFrom [?p ?ref] .
 FILTER NOT EXISTS {?O ?P1 ?S.}
 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}</pre></td></tr>
<tr><td>T4</td><td><pre>G1: Statements where the subject is not an instance of the most used subject class:

SELECT ?S ?SLabel ?O ?OLabel
WHERE {
 ?S wdt:<PropertyID> ?O.
 FILTER NOT EXISTS {
  ?S wdt:P31* wd:<SubjectID>.
 }
SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}</pre></td></tr>
</table>

	G2: Statements where the object is not an instance of the most used object class:
	
	SELECT ?S ?SLabel ?O ?OLabel
	WHERE {
	 ?S wdt:<PropertyID> ?O.
	 FILTER NOT EXISTS {
	  ?O wdt:P31* wd:<ObjectID>.
	 }
	 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
	}

T5	SELECT ?p (COUNT(?p) AS ?count) WHERE {
	 ?S p:<PropertyID> ?statement.
	 ?statement ps:<PropertyID> ?O.
	 ?S wdt:<PropertyID> ?O.
	 ?statement prov:wasDerivedFrom [?p ?ref] .
	}
	GROUP BY ?p
	ORDER BY DESC(?count)

Appendix B: SPARQL queries for the heuristics-based validation of epidemiological counts in Wikidata

Task	SPARQL query

V1	SELECT * WHERE {
	  ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value; pq:P585 ?date].
	  FILTER(YEAR(?date) < 2019)
	  }

V2	SELECT * WHERE {
	  ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value; pq:P459 ?method].
	  FILTER NOT EXISTS {?method wdt:P279* wd:Q177719}
	  }

V3	SELECT * WHERE {
	  ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value; pq:P585 ?datep].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value1; pq:P585 ?date].
	  FILTER(?value > ?value1)
	  FILTER(?datep - ?date = -1)
	  }

V4	SELECT * WHERE {
	  ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value; pq:P585 ?datep].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value1; pq:P585 ?datef].
	  FILTER(?value = ?value1)
	  FILTER(?datep - ?datef = -2)
	  FILTER NOT EXISTS { ?x p:<PropertyID> [ps:<PropertyID> ?value2; pq:P585 ?date].
	      FILTER(?date = ?datep + 1)
	    }
	}

V5	SELECT * WHERE {
	  ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  ?x p:<PropertyID> [ps:<PropertyID> ?value; pq:P585 ?date].
	  FILTER(?value < 0)

<table>
<tr><td></td><td>}</td></tr>
<tr><td>V6</td><td>SELECT * WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:P8049 [ps:P8049 ?h; pq:P585 ?date].
?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
FILTER(?h > ?c)
}</td></tr>
<tr><td>V7</td><td>SELECT * WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:P8011 [ps:P8011 ?t; pq:P585 ?date].
?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
FILTER(?c >= ?t)
}</td></tr>
<tr><td>V8</td><td>SELECT * WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
?x p:P1120 [ps:P1120 ?d; pq:P585 ?date].
FILTER(?c < ?d)
}</td></tr>
<tr><td>V9</td><td>SELECT * WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
?x p:P8010 [ps:P8010 ?r; pq:P585 ?date].
FILTER(?c < ?r)
}</td></tr>
<tr><td>V10</td><td>SELECT ?y ?date ((?count - ?c1) AS ?diff) WHERE {
SELECT ?y ?c1 ?date (SUM(?c) AS ?count) WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:<PropertyID> [ps:<PropertyID> ?c; pq:P585 ?date].
?x wdt:P361 ?y.
?y p:<PropertyID> [ps:<PropertyID> ?c1; pq:P585 ?date].
}
GROUP BY ?y ?c1 ?date
}
ORDER BY DESC(?diff)</td></tr>
</table>

Appendix C: SPARQL queries for the validation of case fatality rate statements in Wikidata

<table>
<tr><th>Task</th><th>SPARQL query</th></tr>
<tr><td>M1</td><td>SELECT * WHERE {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
?x p:P3457 [ps:P3457 ?value; pq:P585 ?date].
FILTER((?value > 1) || (?value < 0))
}</td></tr>
<tr><td>M2</td><td>SELECT ?x ?c ?d ?value ?date (ABS(?value - ?d / ?c) > 0.001 AS ?diff)
WITH {
SELECT ?x {
?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
}
} as %outbreaks
WITH {
SELECT ?x ?value ?date {
INCLUDE %outbreaks.
?x p:P3457 [ps:P3457 ?value; pq:P585 ?date].</td></tr>
</table>

	<pre> }
	} as %casefatalityrates
	WITH {
	  SELECT ?x ?d ?date {
	    INCLUDE %outbreaks.
	    ?x p:P1120 [ps:P1120 ?d; pq:P585 ?date].
	  }
	} as %deaths
	WITH {
	  SELECT ?x ?c ?date {
	    INCLUDE %outbreaks.
	    ?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
	  }
	} as %cases
	WHERE {
	  INCLUDE %casefatalityrates. INCLUDE %deaths. INCLUDE %cases.
	}
	ORDER BY DESC(?diff)</pre>

M3	<pre>SELECT ?x ?c ?d ?date ((?d / ?c) AS ?m)
	WITH {
	  SELECT ?x {
	    ?x p:P31 [ps:P31 wd:Q3241045; pq:P642 wd:Q84263196].
	  }
	} as %outbreaks
	WITH {
	  SELECT ?x ?d ?date {
	    INCLUDE %outbreaks.
	    ?x p:P1120 [ps:P1120 ?d; pq:P585 ?date].
	  }
	} as %deaths
	WITH {
	  SELECT ?x ?c ?date {
	    INCLUDE %outbreaks.
	    ?x p:P1603 [ps:P1603 ?c; pq:P585 ?date].
	  }
	} as %cases
	WHERE {
	  INCLUDE %deaths. INCLUDE %cases.
	  FILTER NOT EXISTS {?x p:P3457 [ps:P3457 ?value; pq:P585 ?date].}
	}</pre>