Adding COVID information to Wikidata

Stanford University - CS 520: What are knowledge graphs?

Andra Waagmeester¹ and Jose Emilio Labra Gayo²

1) Micelio, Antwerp, Belgium | Email: andra@micelio.be, Twitter: @andrawaag
2) University of Oviedo, Spain | Email: labra@uniovi.es, Twitter: @jelabra
A protocol for adding knowledge to Wikidata: aligning resources on human coronaviruses

Andra Waagmeester\textsuperscript{1}, Egon L. Willighagen\textsuperscript{2}, Andrew I. Su\textsuperscript{3}, Martina Kutmon\textsuperscript{2,4}, Jose Emilio Labra Gayo\textsuperscript{5}, Daniel Fernández-Álvarez\textsuperscript{5}, Quentin Groom\textsuperscript{6}, Peter J. Schaap\textsuperscript{7}, Lisa M. Verhagen\textsuperscript{8} and Jasper J. Koehorst\textsuperscript{7,*}

https://doi.org/10.1186/s12915-020-00940-y
Covid-19: six million doses of hydroxychloroquine donated to US despite lack of evidence

_BMJ_ 2020; 368 doi: https://doi.org/10.1136/bmj.m1166 (Published 23 March 2020)

Cite this as: _BMJ_ 2020; 368:m1166
Protocol to add Covid-19 to Wikidata

What we did...

Egon Willighagen @egonwillighagen · Mar 19, 2020
@lubianat, I just noted your work on w.wiki/Kn8 ... I'm going to add the 'protein encoding genes' to go with them #wikidata #SARS_COV_2

Quick update, I asked @andrawaag if their bot can be used to do this instead (I added one manually)

11:51 PM · Mar 19, 2020 · TweetDeck

1 Retweet 2 Quote Tweets 4 Likes

Andra Waagmeester @andrawaag · Mar 20, 2020
Replying to @egonwillighagen and @lubianat
Not yet, but working with you on building that.
The Gene Wiki project, circa 2008

Data imported from structured databases

Summarized knowledge via crowdsourcing
Wikidata is to data as Wikipedia is to text

Wikidata is a collaboratively edited knowledge base operated by the Wikimedia Foundation

- Completely free, even for commercial usage (CC0)
- Anybody can contribute
- Covers all domains of knowledge
- Extensive item history, talk pages, projects, users
- Integration with the semantic web
- High performance query engine (SPARQL)

- Stable! Long term support not dictated by funding cycles
- Actively developed
- Already has large number of active users, editors, contributors!
Wikidata reconciliation

● Schema reconciliation
  ○ Schema extraction
    ▪ Using similar items as templates
    ▪ Wikiprojects
    ▪ sheXer
  ○ Property proposals

● Data reconciliation
  ○ On labels
  ○ On shared identifiers
  ○ IRI mapping
<wikidata-virus-gene> {  
  p:P31  
  p:P279  
  p:P688  
  geneProducts.  
  p:P703  
  Q7187;  
  # Instance of [P31] gene  
  prov:wasDerivedFrom @E265:ncbi-gene-reference OR @E265:ensembl-gene-reference;
  }

<#P31_instance_of_gene> {  
  @P31_instance_of_gene;  
  @P279_subclass_of_gene?;  
  @P688_encodes?;  
  # Zero or one gene products.  
  }

<#P703_found_in_taxon_virus> ;  
  # In which taxonomy and where in that taxonomy this gene is found

  # Identifiers
  p:P351  
  E266:P351_ncbi_gene_id;  
  # Exactly one ncbi gene identifier
  p:P594  
  E266:P594_enssembl_gene_id*;  
  # Zero or more Ensemble gene identifier
  p:P2393  
  E266:P2393_ncbi_locus_tag;  
  # NCBI Locus tag
  }

## Statement details
<#P31_instance_of_gene> {  
  prov:wasDerivedFrom @E265:ncbi-gene-reference OR @E265:ensembl-gene-reference;
  }

<#P279_subclass_of_gene> {  
  p:P279  
  @#gene_types;  
  # Subclass of [P279] gene types <gene_types>
  prov:wasDerivedFrom @E265:ncbi-gene-reference OR @E265:ensembl-gene-reference;
  }

<#P644_genomic_start> {  
  p:P644  
  xsd:string;  
  # genomic start [P644] value
  p:P659  
  E108:sequence_assembly+;  
  # Qualifier indicating the applicable genomic assembly versions.
  prov:wasDerivedFrom @E265:ensembl-gene-reference;
  }

...
Entity schemas and Shape Expressions at Wikidata
RDF and knowledge graphs, the good parts

Data integration

- Merging RDF graphs automatically
- RDF as a basis for knowledge representation

Flexibility

- Data that can be adapted to multiple environments
- Reusable data by default

Tools

- Data stores and SPARQL endpoints
- Multiple serializations: Turtle, JSON-LD, RDF/XML,...
- Embeddable in HTML (Microdata/RDFa)
RDF and knowledge graphs, other parts...

Consuming & producing data from RDF

SPARQL endpoints are usually not well documented

Typical documentation = set of SPARQL queries

Difficult to know where to start doing queries
Why Shape Expressions?

For producers
- Understand the contents they will produce
- Ensure contents have the expected structure
- Advertise and document the structure
- Generate user interfaces

For consumers
- Understand content’s structure
- Validate before processing
- Query generation & optimization
Shape Expressions

Language to describe and validate RDF data

Human readable

Intended audience: domain experts

Syntax inspired by Turtle and SPARQL

Machine processable

Formal semantics

Several syntaxes (Compact, RDF, JSON-LD)

Open source implementations:

- Javascript
- Scala
- Java
- Python

Online demos: RDFShape, ShEx-simple

ShEx schema

```shex
prefix schema: <http://schema.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix : <http://example.org/>

:User IRI {
  schema:name xsd:string ;
  schema:knows @:User * 
}
```

Shape Expressions example

```
prefix schema: <http://schema.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix : <http://example.org/>

:alice schema:name "Alice" ;
   schema:knows :alice, :bob .
:bob schema:name "Robert" .
```

```
prefix schema: <http://schema.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix : <http://example.org/>

:User IRI {
   schema:name xsd:string ;
   schema:knows @:User *
}
```

```
prefix schema: <http://schema.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix : <http://example.org/>

:alice@:User
```

Try it with RDFShape
More info about Shape Expressions

Validating RDF Data book

http://book.validatingrdf.com/

Shapes applications and tools tutorial at ISWC’20

http://www.validatingrdf.com/tutorial/iswc2020/
Shape Expressions at Wikidata

Introduced in May, 2019 as a new namespace: Entity Schemas

- Enn = Entity schemas, e.g. Author: E42

Other namespaces:

- Qnn = Entities, e.g. Douglas Adams: Q42
- Pnn = Properties, e.g. country of citizenship: P27
- Lnn = Lexemes (*lexicographical data*), e.g.: "answer" L42
- Snn = Senses
- ..
Some use cases and tools

Describing and validating entities with entity schemas

Authoring entity schemas

Extracting schemas from existing data: sheXer

Generating User interfaces from entity schemas: shapeForms

Entity schemas ecosystem
Describing and validating entities with entity schemas

Describe expected shape of entities
Check if entities conform to that shape
Filter entities according to conformance

Example: Author (E42)
Authoring and visualizing entity schemas

**YaSHE**: Editor with syntax highlighting, error detection, auto-complete, etc.

**ShEx-Author**: Visual editor

**RDFShape**: RDF playground

  Conversion, Querying, visualization, Validation…)

Can be used to visualize Shape Expressions

**Wikishape**

Similar to RDFShape, but specialized for Wikidata/Wikibase
Extracting schemas from existing data

**sheXer** extracts ShEx schemas from RDF data

- Identify common structure of a given set of items
- Items can be selected by Class, SPARQL query, shapeMap, etc.

Integrated in **RDFShape** and **WikiShape**
Creating UIs from entity schemas

Generate Forms from entity schemas

Suggest/check properties and values

Prototypes

CRADLE
ShapeForms
Entity schemas ecosystem

Increasing adoption of entity schemas at Wikidata

Schemas project: https://www.wikidata.org/wiki/Wikidata:Schemas

Directory of entity schemas

New challenges

Collaborative work and different views

Trade-off: quality of data vs freedom

Domain experts from different disciplines
Community engagement and model discussion
Formally capture and describe model and community consensus

Model development

- Legacy review – develop punch lists for existing data issues that need fixing
- Documentation – terse, human-readable representation helping contributors and maintainers quickly grok the model
- Client pre-submission – submitters test their data before submission to make sure they’re saying what they want to say and that the receiving schema can accommodate all of their data
- Server pre-ingestion – submission process checks data as it comes in and either rejects or warns about non-conformant data
Seeding with data

- Model structure of items (genes, drugs, diseases, .. etc) & relationships between items
- Import data from many sources and ontologies
- Linked to many identifiers from external databases
- Architecture for maintaining data from external sources
A Wikidata Python module integrating the MediaWiki API and the Wikidata SPARQL endpoint

397 commits  2 branches  1 release  7 contributors

sebotic fixed an omission where new items don't get created when domain not s...  22 hours ago

doc Wikidata to Wikipedia mapping prototype for diseases added.  2 years ago

wikidataintegrator fixed an omission where new items don't get created when domain not s...  22 hours ago

Running Bots

<table>
<thead>
<tr>
<th>S</th>
<th>Name</th>
<th>Last Success</th>
<th>Last Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ProteinBot_homo_sapiens</td>
<td>1 day 21 hr - #12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>GOBot_bigmem</td>
<td>2 days 15 hr - #15</td>
<td>9 days 15 hr - #14</td>
</tr>
<tr>
<td></td>
<td>GeneBot_Homo_sapiens</td>
<td>2 days 19 hr - #25</td>
<td>2 days 20 hr - #24</td>
</tr>
<tr>
<td></td>
<td>Disease_Ontology</td>
<td>2 days 23 hr - #11</td>
<td>4 days 13 hr - #8</td>
</tr>
<tr>
<td></td>
<td>GeneDiseaseBot</td>
<td>2 days 23 hr - #9</td>
<td>1 mo 6 days - #2</td>
</tr>
</tbody>
</table>
Feedback loop

diseases

ontologies

genes

proteins

drugs
Corona viruses added to Wikidata

<table>
<thead>
<tr>
<th>Virus strain</th>
<th>NCBI Taxon ID</th>
<th>Wikidata Qid</th>
<th># Genes</th>
<th># Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS virus</td>
<td>694009</td>
<td>Q278567</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>Middle East respiratory syndrome coronavirus</td>
<td>1335626</td>
<td>Q4902157</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>Human coronavirus NL63</td>
<td>277944</td>
<td>Q8351095</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Human coronavirus 229E</td>
<td>11137</td>
<td>Q16983356</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Human coronavirus HKU1</td>
<td>290028</td>
<td>Q16983360</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Human coronavirus OC43</td>
<td>31631</td>
<td>Q16991954</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>SARS-CoV-2</td>
<td>2697049</td>
<td>Q82069695</td>
<td>11</td>
<td>27</td>
</tr>
</tbody>
</table>
Conclusions & future work

Towards an entity schemas ecosystem

Wikidata as a hub for collaborative work

Further work

Apply the protocol to other domains in wikidata

Wikibase related projects

Wikidata subsetting work (see supplementary slides)
Acknowledgments

Maastricht University (NL)
● Egon L. Willighagen
● Martina Kutmon

The Scripps Research Institute (USA)
● Andrew I. Su,

University of Oviedo (ES)
● Daniel Fernández-Álvarez,

Meise Botanic Garden (BE)
● Quentin Groom,

Wageningen University (NL)
● Peter J. Schaap
● Jasper J. Koehorst

Intervacc (NL)
● Lisa M. Verhagen

Gene Wiki

ShEx CG

Wikidata Community

Funders

● National Institute of General Medical Sciences (R01 GM089820)
● Alfred P. Sloan Foundation (grant number G-2019-11458)
● Spanish Ministry of Economy and Competitiveness (Society challenges: TIN2017-88877-R)
● Netherlands Organisation for Scientific Research funded UNLOCK project (NRGWI.obrug.2018.005)
● SYNTHESYS+ a Research and Innovation action funded under H2020-EU.1.4.1.2. Grant agreement ID: 823827.
● ZonMw (grant number: 10430012010015)
End of presentation
Supplementary slides
Wikidata subsetting

Community project in progress at different hackathon’s:

Biohackathon 2020, SWAT4HCLS, …
ShEx Community Group

More info: http://shex.io/

How to join: https://www.w3.org/community/shex/