Introducing Wikidata to the Linked Data Web

Fredo Erxleben[1], Michael Günther[1], Markus Krötzsch[1], Julian Mendez[1], and Denny Vrandečić[2]

[1] Technische Universität Dresden, Germany
[2] Google

Abstract. Wikidata is the central data management platform of Wikipedia. By the efforts of thousands of volunteers, the project has produced a large, open knowledge base with many interesting applications. The data is highly interlinked and connected to many other datasets, but it is also very rich, complex, and not available in RDF. To address this issue, we introduce new RDF exports that connect Wikidata to the Linked Data Web. We explain the data model of Wikidata and discuss its encoding in RDF. Moreover, we introduce several partial exports that provide more selective or simplified views on the data. This includes a class hierarchy and several other types of ontological axioms that we extract from the site. All datasets we discuss here are freely available online and updated regularly.

1 Introduction

Wikidata is the community-created knowledge base of Wikipedia, and the central data management platform for Wikipedia and most of its sister projects [21]. Since its public launch in late 2012, the site has gathered data on more than 15 million entities, including over 34 million statements, and over 80 million labels and descriptions in more than 350 languages. This is the work of well over 40 thousand registered users who have actively contributed so far. Their ceaseless efforts continue to make Wikidata more and more comprehensive and accurate.

One reason for this strong community participation is the tight integration with Wikipedia: as of today, almost every Wikipedia page in every language incorporates content from Wikidata. Primarily, this concerns the links to other languages shown on the left of every page, but Wikipedia editors also make increasing use of the possibility to integrate Wikidata content into articles using special syntax. Upcoming improvements in Wikidata's query capabilities will add more powerful options for doing this, which is expected to further increase the participation of the Wikipedia communities in Wikidata.

A result of these efforts is a knowledge base that is a valuable resource for practitioners and researchers alike. Like Wikipedia, it spans a wide body of general and specialised knowledge that is relevant in many application areas. All of the data is freshly curated by the Wikimedia community, and thus new and original. Naturally, the data is also completely free and open. More than any other factor, however, it is the richness of the data that makes Wikidata unique. Many statements come with provenance information or include additional context data, such as temporal validity; data is strongly connected to external datasets in many domains; and all of the data is multi-lingual by design. Moreover, the data is highly dynamic and based on complex community processes that are interesting in their own right.

Thus, the relevance of Wikidata for researchers in semantic technologies, linked open data, and Web science hardly needs to be argued for. Already the success of DBpedia [2] and Yago(2) [11] testifies to the utility of Wikipedia-related data in these areas. Of course, both projects are fundamentally different from Wikidata in the way they extract data from Wikipedia articles, yet they are similar in scope and scale. Other related projects include Cyc [14] and Freebase [3]. Again, these approaches differ from Wikidata in important aspects [21], but their general scope and fields of application overlap with Wikidata. The wide use of both projects hints at the potential of such efforts.

In spite of all this, Wikidata has hardly been used in the semantic web community so far. The simple reason is that, until recently, the data was not available in RDF. To change this, we have developed RDF encodings for Wikidata, implemented a tool for creating file exports, and set up a site where the results are published. Regular RDF dumps of the data can now be found at http://tools.wmflabs.org/wikidata-exports/rdf/. This paper describes the design underlying these exports and introduces the new datasets:

- In Section 2, we introduce the data model of Wikidata, which governs the structure of the content that we want to export.
- In Section 3, we present the RDF encoding of this content. This includes URI schemes, our handling of multilingual content, and our use of external vocabularies.
- The rich information in Wikidata also leads to relatively complex RDF encodings. Therefore, in Section 4, we discuss alternative and simplified RDF encodings, which we provide for applications that do not require access to all aspects of the data.
- Wikidata also contains interesting schema information which can be expressed naturally using RDFS and OWL. In Section 5, we present several forms of terminological information that we obtain from Wikidata, and for which we provide exports as well.
- The content of Wikidata is strongly connected to external datasets, but rarely uses URIs to do so. To fully integrate Wikidata with the linked data Web, we translate many references to external databases into URIs, as explained in Section 6.
- In Section 7, we present the actual export files and provide some statistics about their current content.

Besides the actual file exports, we also provide all source code that has been used for creating them. Our implementation is part of *Wikidata Toolkit*,[3] a Java library for working with Wikidata content that we develop.

2 The Data Model of Wikidata

Like Wikipedia, Wikidata is organised in pages, and this is also how the data is structured. Every subject on which Wikidata has structured data is called an *entity*, and every entity has a page. The system distinguishes two types of entities so far: *items* and *properties*. In the familiar terms of semantic technologies, items represent individuals and classes, and Wikidata properties resemble RDF properties. Virtually every Wikipedia article in any language has an associated item that represents the subject of this article.

Every item has a page where users can view and enter the data. For example, the item page for the English writer Douglas Adams can be seen at https://www.wikidata.

[3] https://www.mediawiki.org/wiki/Wikidata_Toolkit

Fig. 1. Excerpt of a typical Wikidata item page with terms, statements, and site links

org/wiki/Q42; an excerpt is shown in Fig. 1. The title of this page is "Q42" rather than "Douglas Adams" since Wikidata is a multi-lingual site. Therefore, items are not identified by a label in a specific language, but by an opaque item identifier, which is assigned automatically when creating the item and which cannot be changed later on. Item identifiers always start with "Q" followed by a number. Every item page contains the following main parts:

- the *label* (e.g., "Douglas Adams"),
- a short *description* (e.g., "English writer and humorist"),
- a list of *aliases* (e.g., "Douglas Noël Adams"),
- a list of *statements* (the richest part of the data, explicated below),
- the list of *site links* (links to pages about the item on Wikipedia and other projects).

The first three pieces of data (label, descriptions, aliases) are collectively known as *terms*. They are mainly used to find and to display items. An item can have terms in every language supported by Wikidata. What is displayed on the pages depends on the language setting of the user.

Site links can be given for any of the 286 language editions of Wikipedia, and for several sister projects, such as Wikivoyage and Wikimedia Commons. Site links are functional (at most one link per site) and inverse functional (injective; at most one item for any site link). As opposed to the former system of Wikipedia language links, site links should only be used for articles that are exactly about the item, not about a broader, narrower, or otherwise related topic. Some items do not have any site links, e.g., the item "female" (Q6581072) which is used as a possible value for the sex of persons.

2.1 Properties and Datatypes

Figure 1 shows a simple example statement, which closely resembles an RDF triple with subject *Douglas Adams* (Q42), property *date of birth*, and value 11 March 1952.

Table 1. Wikidata datatypes and their current member fields and field types

Datatype	Member fields
Item	item id (IRI)
String	string
URL	URL (IRI)
Commons Media	article title (string)
Time	point in time (dateTime), timezone offset (int), preferred calendar (IRI), precision (byte), before tolerance (int), after tolerance (int)
Globe coordinates	latitude (decimal), longitude (decimal), globe (IRI), precision (decimal)
Quantity	value (decimal), lower bound (decimal), upper bound (decimal)

Properties, like items, are described on pages and use opaque identifiers starting with "P." For example, *date of birth* is actually P569. Properties do have terms (labels etc.), but no statements or site links.[4]

In addition, Wikidata properties also have a *datatype* that determines the kind of values they accept. The datatype of *date of birth* is *time*. Table 1 (left) shows the list of all available datatypes. Most types are self explaining. *Commons media* is a special type for referring to media files on the Wikimedia Commons media repository used by all Wikipedias. Datatypes determine the structure of the values accepted by properties. A single property value may correspond to a single RDF resource (as for type *item*) or to a single RDF literal (as for type *string*); or it may be a complex value that requires several elements to be described, as for *time*, *globe coordinates*, and *quantity*. Table 1 (right) shows the essential components of each value, as they would appear in RDF.

Many of the member fields should be clear. For *time*, we store an additional timezone offset (in minutes) and a reference to the calendar model that is preferred for display (e.g., Julian calendar, Q1985786); our RDF exports always specify dates in (proleptic) Gregorian calendar. The remaining members of *time* allow to indicate the precision to express uncertain values such as "September 1547" or "3rd century." The details are not essential here. For the most common types of imprecision (precision to day, month, year) we use specific XML Schema datatypes (`xsd:date`, `xsd:gYearMonth`, `xsd:gYear`) to encode this information directly in the literal that specifies the main time point.

For *globe coordinates*, the only unusual member field is *globe*, which gives the celestial body that the coordinates refer to (e.g., Earth, Q2). The remaining members for *globe* and *quantity* are again means of specifying imprecision. Finally, we remark that it is planned to extend quantities with units of measurement in 2014, which will then become another member.

2.2 Complex Statements and References

The full data model of Wikidata statements is slightly more complex than Fig. 1 might suggest. On the one hand, statements can be enriched with so-called *qualifiers*, which provide additional context information for the claim. On the other hand, every statement can include one or more *references*, which support the claim. A statement where both

[4] Support for statements on property pages is under development.

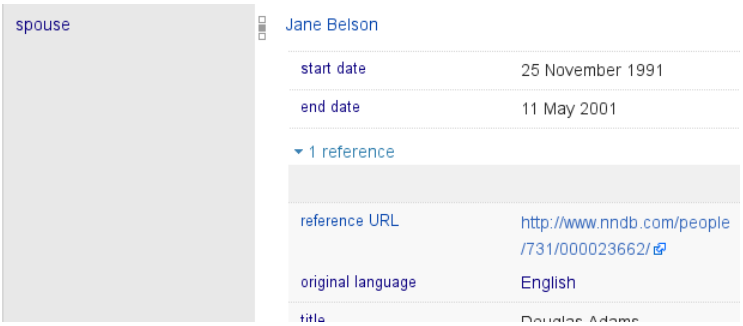

Fig. 2. Part of a complex statement about the wife of Douglas Adams as displayed in Wikidata

aspects are given is shown in Fig. 2. The main property-value pair in this statement is "spouse: Jane Belson" (P26: Q14623681), but there is additional context information.

The qualifiers in Fig. 2 are "start date: 25 November 1991" and "end date: 11 May 2011," which state that Adams has been married to Belson from 1991 till his death in 2011. As before, we are using properties *start date* (P580) and *end date* (P582) of suitable types (time). These property-value pairs refer to the main part of the statement, not to the item on the page (Adams). In RDF, we will need auxiliary nodes that the qualifiers can refer to – the same is true for the references.

Qualifiers are used in several ways in Wikidata. Specifying the validity time of a claim is the most common usage today, so Fig. 2 is fairly typical. However, Wikidata uses many other kinds of annotations that provide contextual information on a statement. Examples include the *taxon author* (P405, essential context for biological taxon names) and the *asteroid taxonomy* (P1016, to contextualise the spectral classification of asteroids). In some cases, qualifiers provide additional arguments of a relationship that has more than two participants. For example, the property *website account on* (P553) specifies a website (such as Twitter, Q918), but is usually used with a qualifier P554 that specifies the account name used by the item on that site. Arguably this is a ternary relationship, but the boundary between context annotation and *n*-ary relation is fuzzy. For example, *Star Trek: The Next Generation* (Q16290) has *cast member* (P161) *Brent Spiner* (Q311453) with two values for qualifier *character role* (P453): *Data* (Q22983) and *Lore* (Q2609295). Note that the same property can be used in multiple qualifiers on the same statement.

The first part of the (single) reference in Fig. 2 is displayed below the qualifiers. Each reference is simply a list of property-value pairs. Wikidata does not provide a more restricted schema for references since the requirements for expressing references are very diverse. References can be classical citations, but also references to websites and datasets, each of which may or may not be represented by an item in Wikidata. In spite of this diversity, references are surprisingly uniform across Wikidata: as of April 2013, there are 23,225,184 pointers to references in Wikidata statements, but only 124,068 different references. This might be unexpected since Wikidata does not support the re-use of references, i.e., the system really stores 23,225,184 lists of property-value pairs which just happen to be the same in many cases. The reason for this uniformity are community

processes (stating how references are to be structured), but also systematic imports that used one source for many statements.

We have used property-value pairs are used in many places: as main parts of claims, as qualifiers, and in references. In each of these cases, Wikidata supports two special "values" for *none* and *some*. *None* is used to say that the given property has no value as in "Elizabeth I of England had no spouse." Similar to negation in OWL, this allows us to state a simple form of negative information to distinguish it from the (frequent) case that information is simply incomplete. This also allows us to add references for negative claims. *Some* is used when we know that a property has a value, but cannot provide further details, as in "Pope Linus had a date of birth, but it is unknown to us." This is similar to the use of blank nodes in RDF and to *someValuesFrom* restrictions in OWL. Formally speaking, neither *none* nor *some* are values that belong to a particular datatype. Nevertheless, both of these special "values" can be used in all places where normal property values are allowed, hence we will usually not mention them explicitly.

2.3 Order and Ranking

All data in Wikidata is ordered – aliases, statements, property-value pairs in a reference, etc. However, representing order is difficult in RDF, since triples in an RDF graph are naturally unordered. Fortunately, the ordering information is only used for presentation, and is not considered meaningful for query answering in Wikidata. It is neither possible nor planned to have queries that can retrieve data based on, e.g., statement order. Hence, we ignore this information in our exports, although this means that the RDF export does not really capture all aspects of the data faithfully.

Even if we do not wish to use statement order in query answering, it can still be necessary to distinguish some statements from the rest. For example, Wikidata contains a lot of historic data with suitable qualifiers, such as the population numbers of cities at different times. Such data has many applications, but a simple query for the population of a city should not return a long list of numbers. To simplify basic filtering of data, Wikidata statements can be given one of three *ranks*: normal (used by default), preferred (used to single out values that are preferred over normal ones), and deprecated (to mark wrong or otherwise unsuitable information that is to be kept in the system for some practical reason). Ranks can be used to reduce the complexity of the dataset to include only the most relevant statements; this is also useful for generating RDF exports.

3 Representing Wikidata Content in RDF

We can now present our primary mapping of Wikidata content to RDF (and, occasionally, OWL). We further discuss this encoding and present simplified approaches in Section 4. Wikidata uses a uniform scheme for URIs of all entities, i.e., items and properties:

http://www.wikidata.org/entity/<*id*>

is the URI for an entity with identifier <*id*>, such as Q42 or P184. These URIs follow linked data standards [1]. They implement content negotiation and redirect to the most

Table 2. Example RDF serialization of terms (top) and sitelinks (bottom) in Turtle

```
<http://www.wikidata.org/entity/Q80>
  a <http://www.wikidata.org/ontology#Item> ;
  <http://www.w3.org/2000/01/rdf-schema#label> "Tim Berners-Lee"@en ;
  <http://schema.org/description> "izumitelj World Wide Weba"@hr ;
  <http://www.w3.org/2004/02/skos/core#altLabel> "TimBL"@pt-br .

<http://es.wikipedia.org/wiki/Tim_Berners-Lee>
  a <http://www.wikidata.org/ontology#Article> ;
  <http://schema.org/about> <http://www.wikidata.org/entity/Q80> ;
  <http://schema.org/inLanguage> "es" .
```

suitable data, which might be an RDF document with basic information about the entity, or the HTML page of the entity on Wikidata. Page URLs are of the form http://www.wikidata.org/wiki/Q42 for items and of the form http://www.wikidata.org/wiki/Property:P184 for properties. In addition, there are specific URLs to obtain the data in several formats, such as http://www.wikidata.org/wiki/Special:EntityData/Q42.nt (RDF in NTriples format) or http://www.wikidata.org/wiki/Special:EntityData/Q42.json (JSON). The RDF information provided by the live export of the site is currently limited to term data; while this already satisfies linked data standards as a basic "useful" piece of information, it is intended to provide all of the data that is found in the dumps there in the future.

In addition to the URIs of Wikidata entities, we also use URIs from an ontology that captures general concepts of the Wikidata data model explained earlier. The base URI of this ontology is http://www.wikidata.org/ontology#, and its current version used for the exports is included in the export directory.

3.1 Exporting Terms and Site Links

We start by describing the RDF export of terms and site links. Labels, descriptions, and aliases can be given in any of the more than 350 languages supported by Wikidata. For exporting this data in RDF, we need to use language tags that follow the BCP 47 standard. We use the tags of the IANA language tag registry[5] whenever possible, although the official tag does not always agree with the identifier used in Wikipedia and Wikidata. For example, als.wikipedia.org is the Alemannic edition of Wikipedia, which has IANA code "gsw" rather than "als" (Tosk Albanian). We translate such exceptions accordingly. Finally, some languages supported by Wikidata do not have their own IANA tag, and we coin a suitable custom tag following the rules of BCP 47. For example, Basa Banyumasan is represented by the tag "jv-x-bms" as an extension to Javanese (jv).

Wikidata terms are then exported as RDF string literals with language tags. We use standard vocabularies for each type of data: RDFS label for labels, schema.org description for descriptions, and SKOS altLabel for aliases. Table 2 (top) shows examples

[5] http://www.iana.org/assignments/language-subtag-registry/

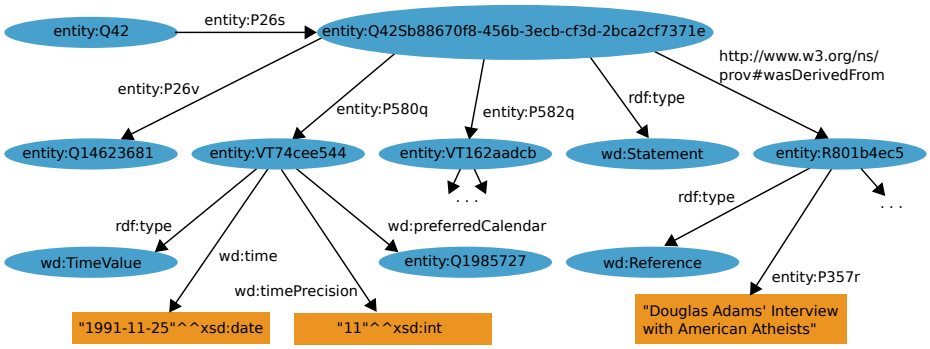

Fig. 3. Partial RDF graph for the statement in Fig. 2

for each. Whenever we use third-party vocabularies, we include an OWL declaration to clarify the type of the property, e.g., DatatypeProperty for description and altLabel.

For 286 languages used in Wikidata, there are also corresponding Wikipedia editions, which might be pointed to in site links. This means that Wikidata contains terms in languages that are not found in any Wikipedia. There are several reasons for this fact. First, Wikipedia editions are only created for languages that have a sufficient community of editors to maintain such a project. This is the reason why languages such as Herero and Afar do not currently have a Wikipedia. Secondly, Wikipedia editions generally try to combine closely related languages. For example, there is only one Portuguese Wikipedia, while Wikidata distinguished Brazilian Portuguese as a separate language that may use different labels. Thirdly, Wikidata may provide terms for the same language in different scripts, such as Kazakh in Arab, Cyrillic, and Latin scripts.

Site links are exported using the schema.org property about to associate a Wikipedia page URL with its Wikidata item, and the schema.org property inLanguage to define the BCP 47 language code of a Wikipedia page. Table 2 (bottom) gives an example.

3.2 Representing Statements and References in RDF

We will now present our approach for modelling Wikidata statements in RDF. A discussion of this modelling and possible alternatives follows in Section 4. The result of modelling the statement of Fig. 2 is displayed in Fig. 3. Qualified names are used for abbreviating URIs, where entity: represents http://www.wikidata.org/entity/, wb: represents http://www.wikidata.org/ontology#, and rdf: and xsd: are as usual. We will explain the parts of this graph step by step.

As discussed in Section 2.2, Wikidata statements are not just triples but can have additional quantifiers and references. The natural approach of representing such data in RDF is to introduce an auxiliary individual that represents the statement itself, denoted by entity:Q42Sb88670f8-456b-3ecb-cf3d-2bca2cf7371e in Fig. 3 (the lengthy identifier is based on a UUID defined by Wikidata for every statement). We can then relate items to statements, and statements to values, qualifier values, and references.

A consequence of this approach is that Wikidata properties do not directly correspond to properties in RDF. A direct relationship as expressed by property "spouse" (P26) in Fig. 2 is broken into two triples, relating item to statement and statement to value, respectively. We use two RDF properties to capture this: entity:P26s to link to the statement and entity:P26v to link to the value (which is entity:Q1985727 in Fig. 3). As discussed in Section 4, using two distinct properties is preferable here.

For qualifiers cannot be annotated further, so we can relate them directly to the statement, without introducing additional resources. To distinguish these triples from the main value, we create another RDF property by appending q to the Wikidata property URI. The qualifiers in Fig. 3 use properties entity:P582q and entity:P580q, respectively. The underlying Wikidata properties are of type *time* in both cases. To express all member fields of this complex datatype shown in Table 1, we introduce additional RDF individuals to represent these values. Figure 3 shows the triples for the value displayed as "25 November 1991" in Fig. 2. The value entity:Q1985727 for the preferred calendar model is the Wikidata item for the proleptic Gregorian calendar.

Values of types *time*, *globe coordinates*, and *quantity* are represented in this fashion, using additional individuals that are named with hash-based URIs. Every such complex value is represented only once in RDF, even if it is used many times throughout the data. Values of datatype *string* are represented by RDF string literals; values of the remaining datatypes *item*, *URL*, and *Commons media* are represented by RDF resources.

References are also represented using dedicated individuals with hash-based names. To relate statements to references, we use the property wasDerivedFrom from the W3C PROV Ontology [13]. Property-value pairs in references are again encoded directly, using yet another variant of properties using with postfix r. Like complex values, references are shared among statements, which saves millions of triples in the RDF dumps.

Finally, the special values *none* and *some* are represented using OWL axioms that state that a property has a value (owl:someValuesFrom) or that this is not the case (owl:complementOf). This use of negation does not usually make the ontology inconsistent, since it refers to the level of statements or references. In particular, it is not contradictory if one statement claims that a property has no value while another gives a value. This might even be desirable to capture conflicting claims of different references.

4 Alternative Ways of Expressing Statements in RDF

The RDF exports discussed in Section 3 are faithful representations of all information of Wikidata that is relevant for query answering. In the case of statements, however, RDF leads to a rather complex representation where information is distributed across many triples. In this section, we discuss alternative approaches and introduce a simplified RDF export format that we provide alongside our main exports.

4.1 Design Principles for Mapping Statements to RDF

We start by explaining the design principles that have guided our RDF encoding of statements in Section 3.2. Our solution makes use of *reification*, the process of encoding complex structures in RDF by introducing new individuals to represent them. We reify statements, complex values, and references. Our main design principles are as follows:

1. Reification: all major structures of the Wikidata data model correspond to resources
2. OWL compatibility: our RDF data can also be read as OWL
3. Strong property typing: all properties used in RDF have a specific range and domain
4. URIs for auxiliary resources: we never use blank nodes for objects of the data model
5. Vocabulary re-use: we make use of third-party vocabulary where suitable

Reification is widely acknowledged as the standard solution of representing complex structures in RDF. The Semantic Web Best Practices group recommends a similar encoding for capturing *n*-ary relations, which is closely related to our task [17], and the W3C standard OWL 2 uses reification to support the annotation of axioms [18]. Such general uses of reification should not be confused with the specific reification vocabulary that RDF provides for triples [6]. This approach has been discussed controversially since it is rather inefficient (using four triples to represent one reified triple), and since it lacks a formal semantics (to relate reified and original triple). The crucial difference to our approach is that there is no "non-reified" RDF structure that we start from: we do not reify RDF triples but Wikidata objects. These objects exist as conceptual entities in the domain that we model. Wikidata statements can even be argued to be the primary subjects that Wikidata collects information about. In this sense, representing Wikidata objects by RDF individuals is not a technical workaround but a modelling goal. We will discuss other approaches that achieve a similar goal later in this section.

OWL compatibility is motivated by our general goal to support the widest range of consumers possible. While OWL can be used on any RDF graph (OWL Full), reasoning on large ontologies is more practical using one of the lightweight profiles of OWL 2, which impose the stricter requirements of OWL DL. Applications of reasoning in the context of Wikidata are conceivable, given that there is already a fair amount of schema information (see Section 5). In addition, we already use various OWL features to encode Wikidata content, and it would be unfortunate if our exports would not be valid OWL.

Strong property typing is simply good modelling practice, following the general guideline of using one URI for one thing. Moreover, it is a prerequisite for obtaining valid OWL DL, where object properties and data properties are strictly separate.

Our use of URIs for resources also follows best practices. Blank nodes add a certain amount of complexity to processing, and their use in OWL DL is subject to some restrictions. The downside of our choice is that we need to coin URIs for complex values and references, which do not have any identifier in the system. This makes it technically difficult to provide useful linked data for the hash-based URIs of these objects. It would be a considerable overhead for Wikidata to keep a global reverse lookup of all such objects that are currently used on some page (recall that references and values are also shared, and thus do not refer to any entity in their URIs). Nevertheless, using blank nodes instead would hardly improve the situation.

Vocabulary re-use is generally encouraged on the semantic web. A special choice that we made for our exports is to use only one vocabulary for each piece of data, even if several would be available. For example, RDFS label is used for labels, but SKOS prefLabel and schema.org name would be just as suitable. The Wikidata linked data

service actually provides label data in each of these, since LOD consumers may expect labels to be given in a specific form.[6]

4.2 Alternatives to Reification

We now discuss alternative options for modelling statements without (explicit) reification. One of the oldest approaches for avoiding reification is to move from triples to *quads* (quadrupels), where the fourth component can be used to attach context information [9]. In relation to RDF, this has first been suggested by Sean B. Palmer in 2001,[7] but very similar ideas have already been proposed in 1993 for the knowledge representation system Loom [15]. Closely related to our work is the use of quads in YAGO2 to model temporal and spatial context information for statements extracted from Wikipedia [11].

RDF 1.1 introduces N-Quads as an official W3C recommendation [7]. While syntactically similar to earlier proposals, the underlying concept there are so-called RDF *datasets*, and the fourth component of quads is interpreted as the identifier for a *named graph*. RDF 1.1 specifies named graphs to have a "minimal semantics" meaning that entailment is not defined for named graphs. It is left to the application to decide which named graphs are to be considered when computing query results or inferences. The SPARQL query language also provides facilities for interacting with named graphs.

Proposals for the semantics of named graphs have been made [8], but did not find their ways into standards. This topic is more closely related to the general discussion of context modelling in semantic web and AI. Notable works in the area include C-OWL [5], Distributed Description Logic [4], TRIPLE [19], and a context-aware semantics proposed by Guha et al. [10]. In spite of these works, there is no standard approach of reasoning over contextualized data today.

We could have used named graphs instead of reification for representing statements. We would still introduce a URI for each statement and use it as the name for a graph that contains the single main triple of the statement. Everything else would remain unchanged. Complex values and references would be reified as before, since named graphs cannot simplify this encoding any further. The main advantage of this approach would be that it keeps the main property-value assignment of each statement in one triple, avoiding the need for joins in query answering. The main disadvantage is that we loose OWL compatibility, and that essential parts of the modelled data are encoded as annotations on graph names, for which no current standard provides any semantics. Nevertheless, there is no harm in providing another variant of the export to those who prefer this view, and we intend to do so in the future (contributions of interested parties are welcome).

Most recently, Nguyen et al. proposed *singleton properties* as yet another approach of annotating RDF statements [16]. Roughly speaking, they combine the fourth component of quads with the predicate URI, to obtain a new property URI that they require to be globally unique. This approach would work for our setting, but, again, the semantic relationship between triples with singleton properties and their annotations are not captured by standard semantics.

---

[6] Considering the idea of linked data as a facilitator for data integration, it would be preferable if vocabulary providers could agree on a single label property.

[7] Discussion with Bijan Parsia and Aaron Swartz: http://chatlogs.planetrdf.com/rdfig/2001-08-10.txt; resulting email: http://lists.w3.org/Archives/Public/www-rdf-logic/2001Aug/0007.html

4.3 Exporting Statements as Triples

Whether we use reification or named graphs, we cannot avoid to use relatively complex RDF graphs if we want to capture the rich structure of Wikidata statements. Yet, it is sometimes desirable to have a simpler view of the data. We therefore provide several secondary data dumps that are not faithful but still meaningful.

The complexity of serialising statements is caused by qualifiers and references. References provide additional information that could be ignored when interpreting statements. In contrast, omitting qualifiers may change the meaning of the statement substantially. Many qualifiers, such as *start date* and *end date*, restrict the validity of a claim to a particular context, and one would obtain wrong or misleading claims when ignoring this information.

To obtain simpler RDF exports, we thus focus on statements without qualifiers and ignore all references. In this situation, we can represent many statements by single RDF triples. This leads to a different RDF graph structure, and we therefore use new RDF properties which use the postfix c. In addition, many complex values can be reduced to their most important member, so that no additional individuals are required. For example, the statement in Fig. 1 can be represented by a single triple

```
entity:Q42  entity:P569c  "1952-03-11"^^xsd:date .
```

We thus obtain an export of *simple statements* which can be combined with the (faithful) exports of terms and site links.

5 Extracting Schema Information from Wikidata

While most of the content of Wikidata is naturally focussed on instances, there is also an interesting and growing amount of schematic information that we provide exports for. On the one hand, this includes an elaborate class hierarchy that is used to classify Wikidata items; on the other hand, we extract an OWL ontology that captures a variety of constraints on the use of properties in Wikidata.

Classification information can be obtained from the Wikidata properties *instance of* (P31) and *subclass of* (P279). The names of these properties suggest a close relationship to rdf:type and rdfs:subClassOf, and it can be seen from the community discussions that these RDF(S) properties have indeed been an important role model for P31 and P279. To extract this information, we apply the approach for exporting simplified statements of Section 4.3. In particular, we ignore statements with qualifiers. This is hardly a restriction for *subclass of*, but there are many cases where *instance of* is used with a temporal annotation to express the that an item has not always been a member of a certain class.

In addition, the Wikidata community has started to formulate *constraints* for the use of properties. For example, a constraint on property *mother* (P25) specifies that all of its values must be instances of *person* (Q215627). This information is used to detect errors in the data, but also to clarify the intended use of properties.

Constraints are not part of the data model of Wikidata, and are in fact completely ignored by the system. Rather, the Wikidata community developed its own way of encoding constraints on the *talk pages* of Wikidata properties. These pages are normal

Table 3. Property constraints in Wikidata and their number of occurrences as of April 2014

Constraint name	Description	Uses
Single value	Property is functional	305
Unique value	Property is inverse functional	290
Symmetric	Property is symmetric	11
Inverse	Specifies the inverse of a property	25
Format	Values must match a given formatting pattern	282
One of	Values must come from a given list of values	60
Existing file	Values must be files on Wikimedia Commons	23
Value type	Values must have some *instance of* or *subclass of* relation	262
Range	Values must be numbers or times in a certain closed interval	53
Target required claim	Values must be items that satisfy further claims	100
Item	Items with this property must also satisfy further claims	436
Type	Items with this property must have some *instance of* or *subclass of* relation	389
Multi value	Items with this property must use it with two or more values	2
Conflicts with	Items with this property must not satisfy certain other claims	8

wiki pages, similar to articles on Wikipedia, where constraints are defined by suitable formatting commands. Constraint violation reports are generated and uploaded to Wikidata by scripts. Table 3 gives an overview of the current constraints with the names used in Wikidata, together with a short explanation of their meaning. Constraints that require something to "satisfy further claims" usually require statements to be given for one or more properties, optionally with specific values. The most general constraints of this kind are *Item* and *Target required claim*.

Many constraints can be expressed in terms of OWL axioms. In contrast to OWL ontologies, constraints are not used for inferring new information (or even inconsistencies) but to detect possible errors. Nevertheless, the schematic information expressed in constraints is still meaningful and the corresponding OWL ontology makes sense as a high-level description of the data. Thus, we extract constraints and provide a dedicated export of the resulting OWL axioms.

The axioms we extract refer to the RDF encoding of Section 3, and only to the the main property of a statement. Currently, the constraints are not applied for qualifiers or references in Wikidata. Clearly, some constraints are difficult or impossible to express in OWL. *Format* can be expressed using a regular expression datatype facet on xsd:string, but few OWL systems support this. *Existing file* expresses a requirement that is not really part of the semantic model we work in. Most other constraints correspond to OWL axioms in a rather direct way. Interestingly, however, neither *Symmetric* nor *Inverse* can be expressed in OWL. While OWL supports symmetric and inverse properties, these apply only to single triples and cannot entail structures like in Fig. 3.

6 Connecting Wikidata to the Linked Data Web

A key characteristic of linked open data is the interconnection of datasets [1]. Wikidata, too, makes many connections to external datasets from many domains, ranging from

Table 4. Overview of dump files of 20th April 2014

File topic	File size	Triples	Content	
instances	16 M	6,169,821	6,169,821	rdf:type relations
taxonomy	336 K	82,076	40,192	rdfs:subclassOf relations
			41,868	OWL classes
simple-statements	300 M	55,925,337	34,146,472	simplified statements
statements	1.8 G	148,513,453	34,282,659	statements
terms	579 M	106,374,085	47,401,512	labels
			8,734,890	aliases
			35,143,663	descriptions
properties	616 K	52,667	1,005	properties
sitelinks	618 M	126,658,004	37,316,300	site links

international authority files, such as ISSN or VIAF, to highly specialised databases such as HURDAT, the database of North Atlantic hurricanes. However, not all of these data sources provide RDF exports or even URIs for their data, and those that do often consider RDF as a secondary service that is provided as one of many export services.

As a consequence, most databases use identifiers that are not URIs, and (at best) provide some scheme of computing URIs from these ids. For example, the Freebase identifier /m/05r5c (Piano) corresponds to the URI http://rdf.freebase.com/ns/m.05r5c, where one has to replace "/" by "." to obtain the local name. Naturally, Wikidata tends to store the identifier, not the URI. The former is usually more concise and readable, but also required in many applications where the identifier plays a role.

Thus, when exporting Wikidata content to RDF, we do not immediately obtain any links to external datasets. To address this problem, we have manually inspected Wikidata properties of type string, and searched for suitable URIs that can be used instead. If possible, we have exported the data using URIs instead of strings. The URI is exported like a Wikidata property value; we never use owl:sameAs to relate external URIs to Wikidata, since this would often not be justified. In some cases, related URIs are available from third parties, but there is no official URI that is endorsed by the owner of the identifier. For example, there are no URIs for *SMILES* ids as used in chemistry, but ChemSpider[8] serves relevant RDF for these identifiers. We have not exported such URIs so far, but we consider to include them in addition the string ids in the future.

Overall, we have found 17 widely used Wikidata properties for which we generate direct links to other RDF datasets. Linked semantic datasets and knowledge bases include Freebase (P646), the Gene Ontology (P686), ChemSpider (661), PubChem (662), several types of entities found in MusicBrainz (P434–P436, P966, P982, P1004), the Virtual International Authority File VIAF (P214) as well as several other national authority files. In total, this allowed for the creation of 2.5 million links to external databases.

Importantly, our main goal is to generate RDF exports that faithfully represent the original data using the language of RDF and linked data properly. We do not aspire to discover links to external datasets that are not already stated explicitly in the data. In particular, we restrict to target datasets for which Wikidata has a property. In some cases,

[8] http://chemspider.com/

suitable properties might be introduced in the future; in other cases, it might be more suitable for third-party datasets to link to Wikidata.

7 RDF Exports

We provide exports in the form of several bz2-compressed N-Triples files that allow users to get access to part of the data without having to donwload all of it. Exports are created once a month, and historic files will remain available. Links to all exports are found at http://tools.wmflabs.org/wikidata-exports/rdf/.

The main RDF export as described in Section 3 is found in four files: *terms* (labels, descriptions, aliases), *statements*, and *sitelinks* contain parts of the item data; *properties* contains all property data (terms and datatypes). In addition, we provide an export of *simple statements* as discussed in Section 4.3, and an export of the class hierarchy (*taxonomy*) and of corresponding rdf:type relations (*instances*) as discussed in Section 5.

The export results for the Wikidata content dump of 20 April 2014 are shown in Table 4, together with some statistics on their size and number of content objects. In total these files cover 15,093,996 items and 1,005 properties. The exported taxonomy turned out to be very interesting since it was built with the semantics of rdfs:subClassOf in mind. This is completely different from Wikipedia's hierarchy of *categories*, which is based on broader/narrower relations [20], as in *Humans* → *Culture* → *Food and drink* → *Meals* → *Breakfast* → *Bed and breakfast* → *Bed and breakfasts in the United States*. Yago(2) reorganizes Wikipedia categories using WordNet [11], and DBpedia integrates Yago's class hierarchy. Yet, many of the over 150,000 classes in DBpedia are still based on English Wikipedia categories, as in *AmericanAcademicsOfJapaneseDescent*. In contrast, Wikidata provides a completely new dataset, which, while certainly far from perfect, is a promising starting point for future research and applications.

8 Conclusions

Wikidata, its content, and the underlying software are under continued development, the outcome of which is hard to foresee. Given the important role that Wikidata plays for Wikipedia, one can be certain that the project will continue to grow in size and quality. Many exciting possibilities of using this data remain to be explored.

Wikidata has had its origins in the Semantic Web community [12], and continues to be inspired by the research and development in this field. With this paper, the results of these efforts are finally available as machine-readable exports. It remains for the community of researchers and practitioners in semantic technologies and linked data to show the added value this can bring.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data: The story so far. Int. J. of Semantic Web and Information Systems 5(3), 1–22 (2009)

2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – A crystallization point for the Web of Data. J. of Web Semantics 7(3), 154–165 (2009)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. 2008 ACM SIGMOD Int. Conf. on Management of Data. pp. 1247–1250. ACM (2008)
4. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. J. of Data Semantics 1, 153–184 (2003)
5. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) Proc. 2nd Int. Semantic Web Conf. (ISWC'03). LNCS, vol. 2870, pp. 164–179. Springer (2003)
6. Brickley, D., Guha, R. (eds.): RDF Schema 1.1. W3C Recommendation (25 February 2014), available at http://www.w3.org/TR/rdf-schema/
7. Carothers, G. (ed.): RDF 1.1 N-Quads: A line-based syntax for RDF datasets. W3C Recommendation (25 February 2014), available at http://www.w3.org/TR/n-quads/
8. Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. J. of Web Semantics 3(4), 247–267 (2005)
9. Cyganiak, R., Harth, A., Hogan, A.: N-Quads: Extending N-Triples with Context. Public draft (2012), available at http://sw.deri.org/2008/07/n-quads/
10. Guha, R.V., McCool, R., Fikes, R.: Contexts for the semantic web. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) Proc. 3rd Int. Semantic Web Conf. (ISWC'04). LNCS, vol. 3298, pp. 32–46. Springer (2004)
11. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artif. Intell., Special Issue on Artificial Intelligence, Wikipedia and Semi-Structured Resources 194, 28–61 (2013)
12. Krötzsch, M., Vrandečić, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. J. of Web Semantics 5(4), 251–261 (2007)
13. Lebo, T., Sahoo, S., McGuinness, D. (eds.): PROV-O: The PROV Ontology. W3C Recommendation (30 April 2013), available at http://www.w3.org/TR/prov-o
14. Lenat, D., Guha, R.V.: Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. Addison-Wesley (1990)
15. MacGregor, R.M.: Representing reified relations in Loom. J. of Experimental and Theoretical Artificial Intelligence 5, 179–183 (1993)
16. Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF reification? Making statements about statements using singleton property. In: Proc. 23rd Int. Conf. on World Wide Web (WWW'14). ACM (2014), to appear
17. Noy, N., Rector, A. (eds.): Defining N-ary Relations on the Semantic Web. W3C Working Group Note (12 April 2006), available at http://www.w3.org/TR/swbp-n-aryRelations/
18. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-overview/
19. Sintek, M., Decker, S.: TRIPLE – a query, inference, and transformation language for the Semantic Web. In: Horrocks, I., Hendler, J.A. (eds.) Proc. 1st Int. Semantic Web Conf. (ISWC'02). Lecture Notes in Computer Science, vol. 2342, pp. 364–378. Springer (2002)
20. Voss, J.: Collaborative thesaurus tagging the Wikipedia way. CoRR abs/cs/0604036 (2006)
21. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. Comm. ACM (2014), to appear