**Abstract.** DBpedia is one of the earliest and most prominent nodes of the Linked Open Data cloud. DBpedia extracts and provides structured data for various crowd-maintained information sources such as over 100 Wikipedia language editions as well as Wikimedia Commons by employing a mature ontology and a stable and thorough Linked Data publishing lifecycle. Wikidata, on the other hand, has recently emerged as a user curated source for structured information which is included in Wikipedia. In this paper, we present how Wikidata is incorporated in the DBpedia eco-system. Enriching DBpedia with structured information from Wikidata provides added value for a number of usage scenarios. We outline those scenarios and describe the structure and conversion process of the DBpediaWikidata (DBW) dataset.

**Keywords:** DBpedia, Wikidata, RDF

**1. Introduction**

In the past decade, several large and open knowledge bases were created. A popular example, DBpedia [6], extracts information from more than one hundred Wikipedia language editions and Wikimedia Commons [9] resulting in several billion facts. A more recent effort, Wikidata [10], is an open knowledge base for building up structured knowledge for re-use across Wikimedia projects.

At the time of writing, both databases grow independently. The Wikidata community is manually curating and growing the Wikidata knowledge base. The data DBpedia extracts from different Wikipedia language editions, and in particular the infoboxes, are constantly growing as well. Although this creates an incorrect perception of rivalry between DBpedia and Wikidata, it is on everyone’s interest to have a common source of truth for encyclopedic knowledge. Currently, it is not always clear if the Wikidata or the Wikipedia community provide more up-to-date information. In addition to the independent growth of DBpedia and Wikidata, there is a number of structural complementarities as well as overlaps with regard to identifiers, structure, schema, curation, publication coverage and data freshness that are analysed throughout this manuscript.

We argue that aligning both knowledge bases in a loosely coupled way would produce an improved resource and render a number of benefits for the end users. Wikidata would have an alternate DBpedia-based view of its data and an additional data distribution channel. End users would have more options for choosing the dataset that fits better in their needs and use cases. Additionally, it would create an indirect connection between the Wikidata and Wikipedia communities that could enable a big range of use cases.

The remainder of this article is organized as follows. Section 2 provides an overview of Wikidata and DBpedia as well as a comparison between the two datasets. Following, Section 3 provides a rational for the design decision that shaped DBW, while Section 4 details the technical details for the conversion process. A description of the dataset is provided in Section 5 fol-
lowed with statistics in Section 6. Access and sustain-
ability options are detailed in Section 7 and Section 8
discusses possible use cases of DBW. Finally, we con-
clude in Section 9.

2. Background

2.1. Wikidata and DBpedia

Wikidata Wikidata is a community-created knowl-
edge base to manage factual information of Wikipedia
and its sister projects operated by the Wikimedia Foun-
dation [10]. In other words, Wikidata’s goal is to be the
central data management platform of Wikipedia. As of
April 2016, Wikidata contains more than 20 million
items and 87 million statements and has more than
6,000 active users. In 2014, an RDF export of Wiki-
data was introduced [2] and recently a few SPARQL
endpoints were made available as external contribu-
tions as well as an official one later on. Wikidata is a
collection of entity pages. There are two types of entity
pages: items and properties. Every item page contains
labels, short description, aliases, statements and site
links. As depicted in Figure 1, each statement consists
of a claim and one or more optional references. Each
claim consists of a property-value pair, and optional
qualifiers. Values are also divided into three types:
no value, unknown value and custom value. The “no
value” marker means that there is certainly no value
for the property, the “unknown value” marker means
that the property has some value, but it is unknown to
us and the “custom value ” which provides a known
value for the property.

DBpedia The semantic extraction of information
from Wikipedia is accomplished using the DBpedia
Information Extraction Framework (DIEF) [6]. DIEF
is able to process input data from several sources pro-
vided by Wikipedia.

The actual extraction is performed by a set of plug-
ggable Extractors, which rely on certain Parsers for
different data types. Since 2011, DIEF is extended to pro-
vide better knowledge coverage for internationalized
content [5] and further eases the integration of differ-
ent Wikipedia language editions as well as Wikimedia
Commons [9].

2.2. Comparison and complementarity

Both knowledge bases overlap as well as com-
plement each other as described in the high-level
overview below.

Identifiers DBpedia uses human-readable Wikipedia
article identifiers to create IRIs for concepts in
each Wikipedia language edition. Wikidata on the
other hand uses language-independent numeric
identifiers.

Structure DBpedia starts with RDF as a base data
model while Wikidata developed its own data
model, which provides better means for capturing
provenance information. Using the Wikidata data
model as a base, different RDF serializations are
possible.

Schema Both schemas of DBpedia and Wikidata are
community curated and multilingual. The DBpe-
dia schema is an ontology based in OWL that
organizes the extracted data and integrates the
different DBpedia language editions. The Wiki-
data schema avoids direct use of RDFS or OWL
terms and redefines many of them, e.g. wkdt:P31
defines a local property similar to rdf:type.
There are attempts to connect Wikidata properties to RDFS/OWL and provide alternative exports of Wikidata data.

**Curation** All DBpedia data is automatically extracted from Wikipedia and is a read-only dataset. Wikipedia editors are, as a side-effect, the actual curators of the DBpedia knowledge base but due to the semi-structured nature of Wikipedia, not all content can be extracted and errors may occur. Wikidata on the other hand has its own direct data curation interface called WikiBase, which is based on the MediaWiki framework.

**Publication** Both DBpedia and Wikidata publish datasets in a number of Linked Data ways, including datasets dumps, dereferenceable URIs and SPARQL endpoints.

**Coverage** DBpedia provides identifiers for all structural components in a Wikipedia language edition. This includes articles, categories, redirects and templates. Wikidata creates common identifiers for concepts that exist in more than one language. For example, not all articles, categories, templates and redirects from a Wikipedia language edition have a Wikidata identifier. On the other hand, Wikidata has more flexible notability criteria and can describe concepts beyond Wikipedia. There has not yet been a thorough qualitative and quantitative comparison in terms of content but the following two studies provide a good comparison overview [3,7].

**Data Freshness** DBpedia is a static, read-only dataset that is updated periodically. An exception is DBpedia Live (available for English, French and German).

On the other hand, Wikidata has a direct editing interface where people can create, update or fix facts instantly. However, there has not yet been a study that compares whether facts entered in Wikidata are more up to date than data entered in Wikipedia (and thus, transitively in DBpedia live).

### 3. Challenges and Design Decisions

In this section we describe the design decisions we took to shape the DBpediaWikidata (DBW) dataset while maximising compatibility, (re)usability and coherency with regard to the existing DBpedia datasets.

**New IRI minting** The most important design decision we had to take was whether to re-use the existing Wikidata IRIs or minting new IRIs in the DBpedia namespace. The decision dates back to 2013, when this project originally started and after lengthy discussions we concluded that minting new URIs was the only viable option. The main reason was the impedance mis-

---

6[http://wikiba.se/](http://wikiba.se/)
7[http://www.mail-archive.com/dbpedia-discussion@lists.sourceforge.net/msg05494.html](http://www.mail-archive.com/dbpedia-discussion@lists.sourceforge.net/msg05494.html)
match between Wikidata data and DBpedia as both projects have minor deviations in conventions. Thus, creating new IRIs allows DBpedia to make local assertions on Wikidata resources without raising too many concerns.

Re-publishing minted IRIs as linked data Since 2007, there has been many tools created by the DBpedia community to explore and exploit DBpedia data through the DBpedia ontology. Although there does not exists any thorough survey, some of these tools are collected on the DBpedia website and we refer the readers to publications related to DBpedia. The decision to publish DBW, enables those tools that are designed to consume the DBpedia ontology to be able to consume the Wikidata data as well. One other main use case for publishing the DBW dataset is the creation of a new fused version of the Wikimedia ecosystem that integrates data from all DBpedia language editions, DBpedia Commons and Wikidata. Normalizing datasets to a common ontology is the first step towards data integration and fusion. Most companies (e.g. Google, Yahoo, Bing, Samsung) keep these datasets hidden; however, our approach is to keep all the DBpedia data open to the community for reuse and feedback.

Ontology design, reification and querying The DBpedia ontology is a crowdsourced ontology developed and maintained since 2006. The DBpedia ontology has reached a stable state where mostly additions and specializations are added in the ontology. At the time of writing, the DBpedia ontology defines 375 datatypes and units. The Wikidata schema on the other hand is quite new and evolving and thus, not as stable. Simple datatype support in Wikidata started from the beginning of the project but units were only introduced at the end of 2015. In addition, Wikidata did not start with RDF as a primary data representation mechanism. There were different RDF serializations of Wikidata data and in particular different reification techniques. For example the RDF we get from content negotiation is still different from the RDF dumps and the announced reification design. For these reasons we chose to use the DBpedia ontology and simple RDF reification. Performance-wise neither reification techniques brings any great advantage and switching to the Wikidata reification scheme would require to duplicate all DBpedia properties.

4. Conversion Process

The DBpedia Information Extraction Framework was greatly refactored to accommodate the extraction of data in Wikidata. The major difference between Wikidata and the other Wikimedia projects DBpedia extracts is that Wikidata uses JSON instead of WikiText to store items.

In addition to some DBpedia provenance extractors that can be used in any MediaWiki export dump, we defined 10 additional Wikidata extractors to export as much knowledge as possible out of Wikidata. These extractors can get labels, aliases, descriptions, different types of sitelinks, references, statements and qualifiers.

For statements we define a RawWikidataExtractor that extracts all available information but uses our reification scheme (cf. Section 5) and the Wikidata properties and the R2RWikidataExtractor that uses a mapping-based approach to map Wikidata statements to the DBpedia ontology. Figure 2 depicts the current DBW extraction architecture.

4.1. Wikidata Property Mappings

In the same way the DBpedia mappings wiki defines infobox to ontology mappings, in the context of this work we define Wikidata property to DBpedia ontology mappings. Wikidata property mappings can be defined both as Schema Mappings and as Value Transformation Mappings. Related approaches have been designed for the migration of Freebase to Wikidata.

4.1.1. Schema Mappings

The DBpedia mappings wiki is a community effort to map Wikipedia infoboxes to the DBpedia ontology and at the same time crowd-source the DBpedia ontology. Mappings between DBpedia properties and Wikidata properties are expressed as owl:equivalentProperty links in the property definition pages, e.g. dbo:birthPlace is equivalent to wkdt:P569. Although Wikidata does not
4.1.2. Value Transformations

The value transformation takes the form of a JSON structure that binds a Wikidata property to one or more value transformation strings. A complete list of the existing value transformation mappings can be found in the DIEF. The value transformation strings that may contain special placeholders in the form of a ‘$’ sign represent transformation functions. If no ‘$’ placeholder is found, the mapping is considered constant. e.g. “P625”: {"rdf:type": "geo: SpatialThing"}. In addition to constant mappings, one can define the following functions:

$1$ replaces the placeholder with the raw Wikidata value. e.g. “P1566”: {"owl:sameAs": "http://sws.geonames.org/$1/"}.

$2$ replaces the placeholder with an escaped value to form a valid MediaWiki title, used when the value is a Wikipedia title and needs proper whitespace escaping. e.g. “P154”: {"logo": "http://commons.wikimedia.org/wiki/Special:FilePath/$2"}.

$getDBpediaClass$ Using the schema class mappings, tries to map the current value to a DBpedia class. This function is used to extract rdf:type and rdfs:subClassOf statement from the respective Wikidata properties. e.g. “P31”: {"rdf:type": "$getDBpediaClass"} “P279”: {"rdfs: subClassOf": "$getDBpediaClass"}

$getLatitude, $getLongitude, $getGeoRss$ Geo-related functions to extract coordinates from values. The following is a complete geo mapping that the extracts geo coordinates similar to the DBpedia coordinates dataset. For every occurrence of the property P625, four triples — one for every mapping — are generated:

---

14 https://www.w3.org/TR/owl2-new-features/#F12:_Punning

15 http://mappings.dbpedia.org/index.php/OntologyClass:Person

16 https://github.com/dbpedia/extraction-framework/blob/master/dump/config.json

17 http://wiki.dbpedia.org/Downloads
Mappings Application The R2RWikidataExtractor merges the schema and value transformation property mappings. For every statement or qualifier it encounters, if mappings for the current Wikidata property exist, it tries to apply them and emit the mapped triples. Statements or qualifiers without mappings are discarded by the R2RWikidataExtractor but captured by the RawWikidataExtractor (cf. Section 5).

4.2. Additions and Post Processing Steps

Besides the basic extraction phase, additional processing steps are added in the workflow.

Type Inferencing In a similar way DBpedia calculates transitive types for every resource, the DBpedia Information Extraction Framework was extended to generate these triples directly at extraction time. As soon as an rdf:type triple is detected from the mappings, we try to identify the related DBpedia class. If a DBpedia class is found, all super types are assigned to a resource.

Transitive Redirects DBpedia already has scripts in place to identify, extract and resolve redirects. After the redirects are extracted, a transitive redirect closure is calculated and applied in all generated datasets by replacing the redirected IRIs to the final ones.

Validation The DBpedia extraction framework already takes care of the correctness of the extracted datatypes during extraction. This is achieved by making sure that the value of every property conforms to the range of that property (i.e. xsd: date). We provide two additional steps of validation. The first step is performed during extraction and checks if the property mappings has a compatible rdfs:range (literal or IRI) with the current value. The rejected triples are stored for feedback to the DBpedia mapping community. The second step is performed in a post-processing step and validates if the type of the object IRI is disjoint with the rdfs:range or the type of the subject disjoint with the rdfs:domain of the property. These inconsistent triples, although they are excluded from the SPARQL endpoint and the Linked Data interface, are offered for download. These violations may originate from logically inconsistent schema mappings or result from different schema modeling between Wikidata and DBpedia.

4.3. IRI Schemes

As mentioned earlier, we decided to generate the RDF datasets under the wikidata.dbpedia.org domain. For example, wkdt:Q42 will be transformed to dw:Q42.

Reification In contrast to Wikidata, simple RDF reification was chosen for the representation of qualifiers. This leads to a simpler design and further reuse of the DBpedia properties. The IRI schemes for the rdf:Statement IRIs follow the same verbose approach from DBpedia to make them easily writable manually by following a specific pattern. If the value is an IRI (Wikidata Item) then for a subject IRI Qs, a property Px and a value IRI Qv the reified statement IRI has the form dw:Qs_Px_Qv. If the value is a Literal then for a subject IRI Qs, a property Px and a Literal value Lv the reified statement IRI has the form dw:Qs_Px_H(Lv,5), where H() is a hash function that takes as argument a string (Lv) and a number to limit the size of the returned hash (5). The hash function in the case of literals is used to create unique IRI and we consider the value ‘5’ big enough to avoid collisions in that value space and keep it short at the same time. The equivalent representation of the Wikidata example in Section 2 is:

```
1 dw:Q42_P26_Q14623681 a rdf:Statement ;
2 rdf:subject dw:Q42 ;
3 rdf:predicate dbo:spouse ;
4 rdf:object dw:Q14623681 ;
5 dbo:startDate "1991-11-25"^^xsd:date ;
6 dbo:endDate "2001-5-11"^^xsd:date ;
```

Listing 4: Simple RDF reification example

DBW does not provide precision. Property definitions exist in the DBpedia ontology.
Listing 5: Example of splitting duplicate claims with different qualifiers using dbo:wikidataSplitIri

IRI Splitting  The Wikidata data model allows multiple identical claims with different qualifiers. In those not so common cases the DBpedia heuristic for IRI readability fails to provide unique IRIs. We create hash-based IRIs for each identical claim and attach them to the original IRI with the dbo:wikidataSplitIri property. At the time of writing, there are 69,662 IRI split triples and Listing 5 provides an example split IRI.

5. Dataset Description

A statistical overview of the DBpedia dataset is provided in Table 1. We extract provenance information, e.g. the MediaWiki page and revision IDs as well as redirects. Aliases, labels, and descriptions are extracted from the related Wikidata item section and are similar to the RDF data Wikidata provides. A difference to Wikidata are the properties we chose to associate aliases and description. Each row in Table 1 is provided as a separate file to ease the consumption of parts of the DBpedia dataset.

Wikidata sitelinks are processed to provide three datasets: 1) owl:sameAs links between DBpedia IRIs and Wikidata IRIs (e.g. dw:Q42 owl:sameAs db-de: Douglas_Adams), the latter is used for the DBpedia releases in order to provide links between the different DBpedia language editions.

Mapped facts are generated from the Wikidata property mappings (cf. Section 4.1). Based on a combination of the predicate and object value of a triple they are split in different datasets. Types, transitive types, geo coordinates, depictions and external owl:sameAs links are separated. The rest of the mapped facts are in the mappings dataset. The rei-
fied mapped facts (R) contains all the mapped facts as reified statements and the mapped qualifiers for these statements (RQ) are provided separately (cf. Listing 4).

Raw facts consist of three datasets that generate triples with DBW IRIs and the original Wikidata properties. The first dataset (raw facts) provides triples for simple statements. The same statements are reified in the second dataset (R) and in the third dataset (RQ) we provide qualifiers linked in the reified statements. Example of the raw datasets can be seen in Listing 4 by replacing the DBpedia properties with the original Wikidata properties. These datasets provide full coverage and, except from the reification design and different namespace, can be seen as equivalent with the WikidataRDF dumps.

Wikidata statement references are extracted in the references dataset using the reified statement resource IRI as subject and the dbo:reference property. Finally, in the mapping and ontology violation datasets we provide triples rejected according to Section 4.2.

6. Dataset Statistics

The statistics we present are based on the Wikidata XML dump from January 2016. We managed to generate a total of 1.4B triples with 188,818,326 unique resources. In Table 1 we provide the number of triples per combined datasets.

Class & property statistics We provide the 5 most popular DBW classes in Table 3. We managed to extract a total of 7.9M typed Things with Agents and Person as the most frequent types. The 5 most frequent mapped properties in simple statements are provided in Table 4 and the most popular mapped properties in qualifiers in Table 5. Wikidata does not have a complete range of value types and date properties are the most frequent at the moment.

Mapping statistics In total, 269 value transformation mappings were defined along with 185 owl:equivalentProperty and 323 owl:equivalentClass schema mappings. Wikidata has 1935 properties (as of January 2016) defined with a total of 81,998,766 occurrences. With the existing mappings we covered 74.21 % of the occurrences.

Redirects In the current dataset we generated 854,578 redirects – including transitive. The number of redirects in Wikidata is small compared to the project size but is also a relatively new project. As the project matures in time the number of redirects will increase and resolving them will have an impact on the resulting data.

Validation According to Table 1, a total of 2.9M errors originated from 11 wrong Wikidata-to-DBpedia schema mappings and 42,541 triples did not pass the ontology validation (cf. Section 4.2).

Access Statistics There were more than 10 million requests to wikidata.dbpedia.org since May 2015 from 28,557 unique IPs as of February 2016 and the daily visitors range from 300 to 2.7K (cf. Figure 3).
7. Access and Sustainability

This dataset is part of the official DBpedia knowledge infrastructure and is published through the regular releases of DBpedia, along with the rest of the DBpedia language editions. The first DBpedia release that included this dataset is DBpedia release 2015-04. DBpedia is a pioneer in adopting and creating best practices for Linked Data and RDF publishing. Thus, being incorporated into the DBpedia publishing workflow guarantees: a) long-term availability through the DBpedia Association and b) agility in adopting any new best-practices promoted by DBpedia. In addition to the regular and stable releases of DBpedia we provide additional dataset updates from the project website.

Besides the stable dump availability we created http://wikidata.dbpedia.org for the provision of a Linked Data interface and a SPARQL Endpoint. The dataset is registered in DataHub and provides machine-readable metadata as void and DataID [1]. Since the project is now part of the official DBpedia Information Extraction Framework, our dataset reuses the existing user and developer support infrastructure. DBpedia has a general discussion and developer list as well as an issue tracker\textsuperscript{21} for submitting bugs.

8. Use Cases

Although it is early to identify a big range of possible use cases for DBW, our main motivation was a) familiar querying for the DBpedia community, b) vertical integration with the existing DBpedia infrastructure and c) data integration and fusion.

Listings 6 and 7 provide query examples with simple and reified statements. Since DBpedia provides transitive types directly, queries such as \textit{select all places} can be formulated in SPARQL endpoints without SPARQL 1.1 support or simple scripts on the dumps. Moreover, \texttt{dbo:country} can be more intuitive than \texttt{wkdt:P17c}. Finally, the DBpedia queries can, in most cases directly or with minor adjustments, run on all DBpedia language endpoints. This, among others, means that existing DBpedia applications are potentially compatible with DBW. When someone is working with reified statements, the DBpedia IRIs encode all possible information to visually identify the resources and items involved (cf. Section 4.3) in the statement while Wikidata uses a hash string. Querying for reified statement in Wikidata needs to properly suffix the Wikidata property with \texttt{c/s/q}.\textsuperscript{22} Simple RDF reification on the other hand limits the use of SPARQL property path expressions.

The fact that the datasets are split according to the information they contain eases data consumption when someone needs a specific subset, e.g. coordinates. An additional important use case is data integration. Converting a dataset to a common schema, facilitates the integration of data. The DBW dataset is planned to be used as an enrichment dataset on top of DBpedia and fill in data that are being moved from Wikipedia infoboxes to Wikidata. It is also part of our short-term plan to fuse all DBpedia data into a new single knowledge base and the DBW dataset will have a prominent role in this project.

\textit{Use cases for Wikidata} Through DBW Wikidata has a gateway to DBpedia data from other language editions by using the \textit{Wikidata property mappings}. By accessing DBpedia data, Wikidata can cross-reference facts as well as identify & consume data updates from Wikipedia. Another core feature of Wikidata is adding references for each statement. Unfortunately there are many facts copied from Wikipedia by Wikidata editors\textsuperscript{22}.

\textsuperscript{19}https://www.weblogexpert.com/

\textsuperscript{20}http://wikidata.dbpedia.org/report/report.html

\textsuperscript{21}https://github.com/dbpedia/extraction-framework/issues

\textsuperscript{22}At the time of writing, there is a mismatch of the actual Wikidata syntax reported from the Wikidata paper, the Wikidata RDF dumps and the official Wikidata SPARQL endpoint
Table 7  
Technical details of DBW dataset

<table>
<thead>
<tr>
<th>Name</th>
<th>DBW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparql Endpoint</td>
<td><a href="http://wikidata.dbpedia.org/sparql">http://wikidata.dbpedia.org/sparql</a></td>
</tr>
<tr>
<td>Example resource link</td>
<td><a href="http://wikidata.dbpedia.org/resource/Q42">http://wikidata.dbpedia.org/resource/Q42</a></td>
</tr>
<tr>
<td>Download link</td>
<td><a href="http://wikidata.dbpedia.org/downloads">http://wikidata.dbpedia.org/downloads</a></td>
</tr>
<tr>
<td>DataHub link</td>
<td><a href="http://datahub.io/dataset/dbpedia-wikidata">http://datahub.io/dataset/dbpedia-wikidata</a></td>
</tr>
<tr>
<td>Void link</td>
<td><a href="http://wikidata.dbpedia.org/downloads/void.ttl">http://wikidata.dbpedia.org/downloads/void.ttl</a></td>
</tr>
<tr>
<td>DataID link</td>
<td><a href="http://wikidata.dbpedia.org/downloads/20150330/dataid.ttl">http://wikidata.dbpedia.org/downloads/20150330/dataid.ttl</a></td>
</tr>
<tr>
<td>Licence</td>
<td>CC0</td>
</tr>
</tbody>
</table>

Listing 6: Queries with simple statement

```
#DBw
SELECT * WHERE {
?place a dbo:Place ;
dbo:country dw:Q183.
OPTIONAL {
FILTER (LANG(?label)="en")
}
}
#Wikidata
SELECT * WHERE {
?place wdt:P31/wdt:P279+ wd:Q486972;
wdt:P17 wd:Q183.
OPTIONAL {
FILTER (LANG(?label)="en")
}
}
```

Listing 7: Queries with reified statements

```
#DBw
SELECT DISTINCT ?person WHERE {
?statementUri rdf:subject ?person ;
rdf:predicate dbo:spouse ;
dbo:startDate ?m_date.
FILTER (year(?m_date)<2000)
}
#Wikidata
SELECT DISTINCT ?person WHERE {
FILTER (year(?m_date)<2000)
}
```

that cite Wikipedia and not the possible external or authoritative source that is cited in the Wikipedia article. DBpedia recently started extracting citations from Wikipedia pages. This makes it possible to associate facts extracted from DBpedia with citations close to the fact position. By using the DBpedia citations, DBpedia facts and their associations as well as the Wikidata property mappings, a lot of references with high confidence can be suggested for Wikidata facts.

**Combination of both datasets**  
Currently there is indeed an overlap of facts that exist both in DBpedia and Wikidata however, there are also a lot of facts that are unique to each dataset. For instance, DBpedia captures the links between Wikipedia pages that are used to compute page-rank datasets for different Wikipedia/DBpedia language editions. Using the page links from DBpedia and Wikidata as an article association hub, a global page-rank score for Wikidata items that takes the interconnection graph of all Wikipedias is possible. In general DBW provides a bridge that we hope will make it easier for the huge amount of information on both datasets to be used in some new interesting ways and improve Wikidata and Wikipedia itself.

9. Conclusions and Future Work

We present an effort to provide an alternative RDF representation of Wikidata. Our work involved the creation of 10 new DBpedia extractors, a Wikidata2DBpedia mapping language and additional post-processing & validation steps. With the current mapping status we managed to generate over 1.4 billion RDF triples with CC0 license. According to the web server statistics, the daily number of DBW visitors

[^23]: http://wiki.dbpedia.org/about
range from 300 to 2,700 and we counted almost 30,000 unique IPs since the start of the project, which indicates that this dataset is used. In the future we plan to extend the mapping coverage as well as extend the language with new mapping functions and more advanced mapping definitions. The dataset is already part of the bi-yearly DBpedia release cycle and thus regularly updated. We will additionally consider providing DBW as a live service similar to DBpedia Live.

Acknowledgements

This work was funded by grants from the EU’s 7th & H2020 Programmes for projects ALIGNED (GA 644055), GeoKnow (GA 318159) and HOBBIT (GA 688227).

References