DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Jens Lehmann [a,*], Robert Isele [g], Max Jakob [e], Anja Jentzsch [d], Dimitris Kontokostas [a], Pablo N. Mendes [f], Sebastian Hellmann [a], Mohamed Morsey [a], Patrick van Kleef [c], Sören Auer [a], Christian Bizer [b]

[a] *University of Leipzig, Institute of Computer Science, AKSW Group, Augustusplatz 10, D-04009 Leipzig, Germany*
E-mail: {lastname}@informatik.uni-leipzig.de
[b] *University of Mannheim, Research Group Data and Web Science, B6-26, D-68159 Mannheim*
E-mail: chris@informatik.uni-mannheim.de
[c] *OpenLink Software, 10 Burlington Mall Road, Suite 265, Burlington, MA 01803, U.S.A.*
E-mail: pkleef@openlinksw.com
[d] *Hasso-Plattner-Institute for IT-Systems Engineering, Prof.-Dr.- Helmert-Str. 2-3, D-14482 Potsdam, Germany*
E-mail: mail@anjajentzsch.de
[e] *Neofonie GmbH, Robert-Koch-Platz 4, D-10115 Berlin, Germany*
E-mail: max.jakob@neofonie.de
[f] *Kno.e.sis - Ohio Center of Excellence in Knowledge-enabled Computing, Wright State University, Dayton, USA.*
E-Mail: pablo@knoesis.org
[g] *Brox IT-Solutions GmbH, An der Breiten Wiese 9, D-30625 Hannover, Germany*
E-Mail: mail@robertisele.com

Abstract. The DBpedia community project extracts structured, multilingual knowledge from Wikipedia and makes it freely available using Semantic Web and Linked Data standards. The extracted knowledge, comprising more than 1.8 billion facts, is structured according to an ontology maintained by the community. The knowledge is obtained from different Wikipedia language editions, thus covering more than 100 languages, and mapped to the community ontology. The resulting data sets are linked to more than 30 other data sets in the Linked Open Data (LOD) cloud. The DBpedia project was started in 2006 and has meanwhile attracted large interest in research and practice. Being a central part of the LOD cloud, it serves as a connection hub for other data sets. For the research community, DBpedia provides a testbed serving real world data spanning many domains and languages. Due to the continuous growth of Wikipedia, DBpedia also provides an increasing added value for data acquisition, re-use and integration tasks within organisations. In this system report, we give an overview over the DBpedia community project, including its architecture, technical implementation, maintenance, internationalisation, usage statistics and showcase some popular DBpedia applications.

Keywords: Linked Open Data, Knowledge Extraction, Wikipedia, Data Web, RDF, OWL

[*] Corresponding author. E-mail: lehmannn@informatik.uni-leipzig.de.

1. Introduction

The DBpedia community project extracts knowledge from Wikipedia and makes it widely available via established Semantic Web standards and Linked Data best practices. Wikipedia is currently the 7th most popular website[1], the most widely used encyclopedia, and one of the finest examples of truly collaboratively created content. However, due to the lack of the exploitation of the inherent structure of Wikipedia articles, Wikipedia itself only offers very limited querying and search capabilities. For instance, it is difficult to find all rivers that flow into the Rhine or all Italian composers from the 18th century. One of the goals of the DBpedia project is to provide those querying and search capabilities to a wide community by extracting structured data from Wikipedia which can then be used for answering expressive queries such as the ones outlined above.

The DBpedia project was started in 2006 and has meanwhile attracted significant interest in research and practice. It has been a key factor for the success of the Linked Open Data initiative and serves as an interlinking hub for other data sets (see Section 5). For the research community, DBpedia provides a testbed serving real data spanning various domains and more than 100 language editions. Numerous applications, algorithms and tools have been build around or applied to DBpedia. Due to the continuous growth of Wikipedia and improvements in DBpedia, the extracted data provides an increasing added value for data acquisition, re-use and integration tasks within organisations. While the quality of extracted data is unlikely to reach the quality of completely manually curated data sources, it can be applied to some enterprise information integration use cases and has shown to be relevant beyond research projects as we will describe in in Section 7.

One of the reasons why DBpedia's data quality has improved over the past years is that the structure of the knowledge in DBpedia itself is meanwhile maintained by its community. Most importantly, the community creates mappings from Wikipedia information representation structures to the DBpedia ontology. This ontology unifies different template structures, which will later be explained in detail – both within single Wikipedia language editions and across currently 27 different languages. The maintenance of different language editions of DBpedia is spread across a number of organisations. Each organisation is responsible for the support of a certain language. The local DBpedia chapters are coordinated by the DBpedia Internationalisation Committee. In addition to multilingual support, DBpedia also provides data-level links into more than 30 external data sets, which are partially also contributed from partners beyond the core project team.

The aim of this system report is to provide a description of the DBpedia community project, including the architecture of the DBpedia extraction framework, its technical implementation, maintenance, internationalisation, usage statistics as well as showcasing some popular DBpedia applications. This system report is a comprehensive update and extension of previous project descriptions in [1] and [5]. The main novelties compared to these articles are:

- The concept and implementation of the extraction based on a community-curated DBpedia ontology.
- The wide internationalisation of DBpedia.
- A live synchronisation module which processes updates in Wikipedia as well as the DBpedia ontology and allows third parties to keep their copies of DBpedia up-to-date.
- A description of the maintenance of public DBpedia services and statistics about their usage.
- An increased number of interlinked data sets which can be used to further enrich the content of DBpedia.
- The discussion and summary of novel third party applications of DBpedia.

In essence, the report summarizes major developments in DBpedia in the past four years since the publication of [5].

The system report is structured as follows: In the next section, we describe the DBpedia extraction framework, which forms the technical core of DBpedia. This is followed by an explanation of the community-curated DBpedia ontology with a focus on its evolution over the past years and multilingual support. In Section 4, we explicate how DBpedia is synchronised with Wikipedia with just very short delays and how updates are propagated to DBpedia mirrors employing the *DBpedia Live* system. Subsequently, we give an overview of the external data sets that are interlinked from DBpedia or that set data-level links pointing at DBpedia themselves (Section 5). In Section 6, we provide statistics on the access of DBpedia and describe lessons learned for the maintenance of a large scale public data set. Within Section 7, we briefly

[1] See `http://www.alexa.com/topsites`. Retrieved in June 2013.

describe several use cases and applications of DBpedia in a variety of different areas. Finally, we report on related work in Section 8 and conclude in Section 9.

2. Extraction Framework

Wikipedia articles consist mostly of free text, but also comprise various types of structured information in the form of wiki markup. Such information includes infobox templates, categorisation information, images, geo-coordinates, links to external web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia. The DBpedia extraction framework extracts this structured information from Wikipedia and turns it into a rich knowledge base. In this section, we give an overview of the DBpedia knowledge extraction framework.

2.1. *General Architecture*

Figure 1 shows an overview of the technical framework. The DBpedia extraction is structured into four phases:

Input: Wikipedia pages are read from an external source. Pages can either be read from a Wikipedia dump or directly fetched from a MediaWiki installation using the MediaWiki API.

Parsing: Each Wikipedia page is parsed by the wiki parser. The wiki parser transforms the source code of a Wikipedia page into an Abstract Syntax Tree.

Extraction: The Abstract Syntax Tree of each Wikipedia page is forwarded to the extractors. DBpedia offers extractors for many different purposes, for instance, to extract labels, abstracts or geographical coordinates. Each extractor consumes an Abstract Syntax Tree and yields a set of RDF statements.

Output: The collected RDF statements are written to a sink. Different formats, such as N-Triples are supported.

2.2. *Extractors*

The DBpedia extraction framework employs various extractors for translating different parts of Wikipedia pages to RDF statements. A list of all available extractors is shown in Table 1. DBpedia extractors can be divided into four categories:

Mapping-Based Infobox Extraction: The *mapping-based infobox extraction* uses manually written mappings that relate infoboxes in Wikipedia to terms in the DBpedia ontology. The mappings also specify a datatype for each infobox property and thus help the extraction framework to produce high quality data. The mapping-based extraction will be described in detail in Section 2.4.

Raw Infobox Extraction: The *raw infobox extraction* provides a direct mapping from infoboxes in Wikipedia to RDF. As the raw infobox extraction does not rely on explicit extraction knowledge in the form of mappings, the quality of the extracted data is lower. The raw infobox data is useful, if a specific infobox has not been mapped yet and thus is not available in the mapping-based extraction.

Feature Extraction: The *feature extraction* uses a number of extractors that are specialized in extracting a single feature from an article, such as a label or geographic coordinates.

Statistical Extraction: Some NLP related extractors aggregate data from all Wikipedia pages in order to provide data that is based on *statistical measures* of page links or word counts, as further described in Section 2.6.

2.3. *Raw Infobox Extraction*

The type of Wikipedia content that is most valuable for the DBpedia extraction are infoboxes. Infoboxes are frequently used to list an article's most relevant facts as a table of attribute-value pairs on the top right-hand side of the Wikipedia page (for right-to-left languages on the top left-hand side). Infoboxes that appear in a Wikipedia article are based on a template that specifies a list of attributes that can form the infobox. A wide range of infobox templates are used in Wikipedia. Common examples are templates for infoboxes that describe persons, organisations or automobiles. As Wikipedia's infobox template system has evolved over time, different communities of Wikipedia editors use different templates to describe the same type of things (e.g. `Infobox_city_japan`, `Infobox_swiss_town` and `Infobox_town_de`). In addition, different templates use different names for the same attribute (e.g. `birthplace` and `placeofbirth`). As many Wikipedia editors do not strictly follow the recommendations given on the page that describes a template, attribute values are expressed using a wide range of different formats and units of measurement. An excerpt of an infobox that is based on a template for describing automobiles is shown below:

Name	*Description*	*Example*
abstract	Extracts the first lines of the Wikipedia article.	`dbr:Berlin dbo:abstract "Berlin is the capital city of (...)"`
article categories	Extracts the categorization of the article.	`dbr:Oliver_Twist dc:subject dbr:Category:English_novels`
category label	Extracts labels for categories.	`dbr:Category:English_novels rdfs:label "English novels"`
disambiguation	Extracts disambiguation links.	`dbr:Alien dbo:wikiPageDisambiguates dbr:Alien_(film)`
external links	Extracts links to external web pages related to the concept.	`dbr:Animal_Farm dbo:wikiPageExternalLink <http://books.google.com/?id=RBGmrDnBs8UC>`
geo coordinates	Extracts geo-coordinates.	`dbr:Berlin georss:point "52.5006 13.3989"`
grammatical gender	Extracts grammatical genders for persons.	`dbr:Abraham_Lincoln foaf:gender "male"`
homepage	Extracts links to the official homepage of an instance.	`dbr:Alabama foaf:homepage <http://alabama.gov/>`
image	Extracts the first image of a Wikipedia page.	`dbr:Berlin foaf:depiction <http://.../Overview_Berlin.jpg>`
infobox	Extracts all properties from all infoboxes.	`dbr:Animal_Farm dbo:date "March 2010"`
interlanguage	Extracts interwiki links.	`dbr:Albedo dbo:wikiPageInterLanguageLink dbr:Albedo .`
label	Extracts the article title as label.	`dbr:Berlin rdfs:label "Berlin"`

lexicalizations	Extracts information about surface forms and their association with concepts (only N-Quad format).	`dbr:Pine sptl:lexicalization lx:pine_tree ls:Pine_pine_tree`
		`lx:pine_tree rdfs:label "pine tree" .`
		`ls:Pine_pine_tree sptl:pUriGivenSf "0.941" .`

mappings	Extraction based on mappings of Wikipedia infoboxes to the DBpedia ontology.	`dbr:Berlin dbo:country dbr:Germany`
page ID	Extracts page ids of articles.	`dbr:Autism dbo:wikiPageID "25"`
page links	Extracts all links between Wikipedia articles.	`dbr:Autism dbo:wikiPageWikiLink dbr:Human_brain`
persondata	Extracts information about persons represented using the PersonData template.	`dbr:Andre_Agassi foaf:birthDate "1970-04-29"`
PND	Extracts PND (Personennamendatei) data about a person.	`dbr:William_Shakespeare dbo:individualisedPnd "118613723"`
redirects	Extracts redirect links between articles in Wikipedia.	`dbr:ArtificalLanguages dbo:wikiPageRedirects dbr:Constructed_language`
revision ID	Extracts the revision ID of the Wikipedia article.	`dbr:Autism <http://www.w3.org/ns/prov#wasDerivedFrom> <http://en.wikipedia.org/wiki/Autism?oldid=495234324>`
SKOS categories	Extracts information about which concept is a category and how categories are related using the SKOS Vocabulary.	`dbr:Category:World_War_II skos:broader dbr:Category:Modern_history`
thematic concept	Extracts ‘thematic’ concepts, the centers of discussion for categories.	`dbr:Category:Music skos:subject dbr:Music`
topic signatures	Extracts topic signatures.	`dbr:Alkane sptl:topicSignature "carbon alkanes atoms"`
wiki page	Extracts links to corresponding articles in Wikipedia.	`dbr:AnAmericanInParis foaf:isPrimaryTopicOf <http://en.wikipedia.org/wiki/AnAmericanInParis>`

Table 1

Overview of the DBpedia extractors. *sptl:* is used as prefix for `http://spotlight.dbpedia.org/vocab/`, *lx:* is used for `http://spotlight.dbpedia.org/lexicalizations/` and *ls:* is used for `http://spotlight.dbpedia.org/scores/`.

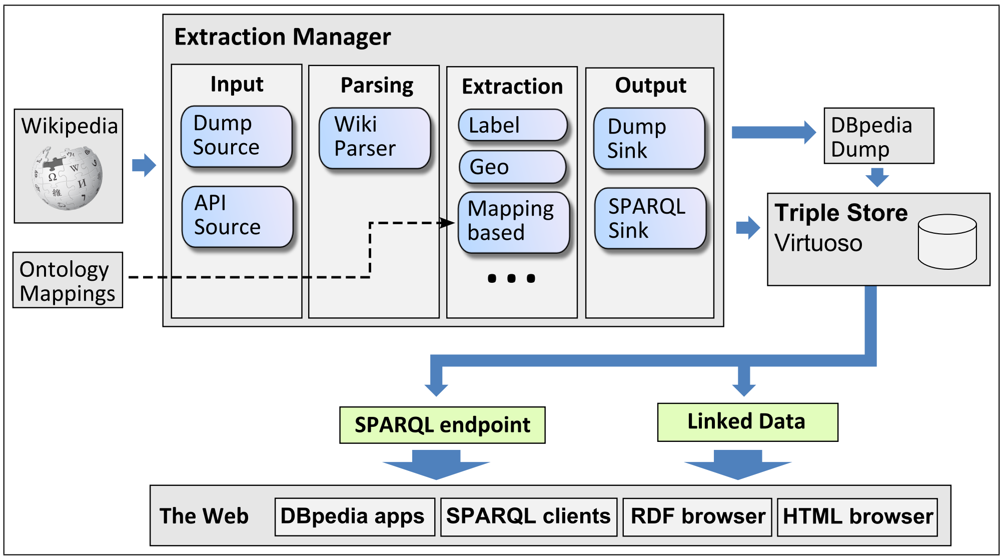


Fig. 1. Overview of DBpedia extraction framework.

```
{{Infobox automobile
| name          = Ford GT40
| manufacturer = [[Ford Advanced Vehicles]]
| production    = 1964-1969
| engine        = 4181cc
(...)
}}
```

In this infobox, the first line specifies the infobox type and the subsequent lines specify various attributes of the described entity.

An excerpt of the extracted data is as follows:[2]

```
dbr:Ford_GT40 [
  dbp:name "Ford GT40"@en;
  dbp:manufacturer dbr:Ford_Advanced_Vehicles;
  dbp:engine 4181;
  dbp:production 1964;
  (...)
] .
```

This extraction output has weaknesses: The resource is not associated to a class in the ontology and the engine and production data use literal values for which the semantics are not obvious. Those problems can be overcome by the mapping-based infobox extraction presented in the next subsection.

2.4. Mapping-Based Infobox Extraction

In order to homogenize the description of information in the knowledge base, in 2010 a community effort has been initiated to develop an ontology schema and mappings from Wikipedia infobox properties to this ontology. The alignment between Wikipedia infoboxes and the ontology is performed via community-provided mappings that help to normalize name variations in properties and classes. Heterogeneity in the Wikipedia infobox system, like using different infoboxes for the same type of entity or using different property names for the same property (cf. Section 2.3), can be alleviated in this way. This significantly increases the quality of the raw Wikipedia infobox data by typing resources, merging name variations and assigning specific datatypes to the values.

This effort is realized using the DBpedia Mappings Wiki[3], a MediaWiki installation set up to enable users to collaboratively create and edit mappings. These mappings are specified using the DBpedia Mapping Language. The mapping language makes use of MediaWiki templates that define DBpedia ontology classes and properties as well as template/table to ontology mappings. A mapping assigns a type from the DBpedia ontology to the entities that are described by the corresponding infobox. In addition, attributes in the infobox are mapped to properties in the DBpedia ontology. In the following, we show a mapping that maps infoboxes that use the `Infobox_automobile` template to the DBpedia ontology:

[2] We use `dbr` for `http://dbpedia.org/resource/`, `dbo` for `http://dbpedia.org/ontology/` and `dbp` for `http://dbpedia.org/property/` as prefixes throughout the report.

[3] `http://mappings.dbpedia.org`

```
{{TemplateMapping
|mapToClass = Automobile
|mappings =
 {{PropertyMapping
 | templateProperty = name
 | ontologyProperty = foaf:name }}
 {{PropertyMapping
 | templateProperty = manufacturer
 | ontologyProperty = manufacturer }}
 {{DateIntervalMapping
 | templateProperty = production
 | startDateOntologyProperty = productionStartDate
 | endDateOntologyProperty = productionEndDate }}
 {{IntermediateNodeMapping
 | nodeClass = AutomobileEngine
 | correspondingProperty = engine
 | mappings =
    {{PropertyMapping
    | templateProperty = engine
    | ontologyProperty = displacement
    | unit = Volume }}
    {{PropertyMapping
    | templateProperty = engine
    | ontologyProperty = powerOutput
    | unit = Power }}
 }}
 (...)
}}
```

The RDF statements that are extracted from the previous infobox example are shown below. As we can see, the production period is correctly split into a start year and an end year and the engine is represented by a distinct RDF node.

```
dbr:Ford_GT40 [
  rdf:type   dbo:Automobile;
  rdfs:label "Ford GT40"@en;
  dbo:manufacturer
             dbr:Ford_Advanced_Vehicles;
  dbo:productionStartYear
             "1964"^^xsd:gYear;
  dbo:productionEndYear "1969"^^xsd:gYear;
  dbo:engine [
             rdf:type AutomobileEngine;
             dbo:displacement "0.004181";
  ]
  (...)
] .
```

The DBpedia Mapping Wiki is not only used to map different templates within a single language edition of Wikipedia to the DBpedia ontology, but is used to map templates from all Wikipedia language editions to the shared DBpedia ontology. Figure 2 shows how the infobox properties *author* and $\sigma\upsilon\gamma\gamma\varrho\alpha\varphi\varepsilon\alpha\varsigma$ – author in Greek – are both being mapped to the global identifier `dbo:author`. That means, in turn, that information from all language versions of DBpedia can be merged and DBpedias for smaller languages can be augmented with knowledge from larger DBpedias such as the English edition. Conversely, the larger DBpedia editions can benefit from more specialized knowledge from localized editions, such as data about smaller towns which is often only present in the corresponding language edition [40].

Besides hosting of the mappings and DBpedia ontology definition, the DBpedia Mappings Wiki offers various tools which support users in their work:

- **Mapping Validator** When editing a mapping, the mapping can be directly validated by a button on the edit page. This validates changes before saving them for syntactic correctness and highlights inconsistencies such as missing property definitions.
- **Extraction Tester** The extraction tester linked on each mapping page tests a mapping against a set of example Wikipedia pages. This gives direct feedback about whether a mapping works and how the resulting data is structured.
- **Mapping Tool** The DBpedia Mapping Tool is a graphical user interface that supports users to create and edit mappings.

2.5. URI Schemes

For every Wikipedia article, the framework introduces a number of URIs to represent the concepts described on a particular page. Up to 2011, DBpedia published URIs only under the `http://dbpedia.org` domain. The main namespaces were:

- *`http://dbpedia.org/resource/`* (prefix *dbr*) for representing article data. Apart from a few corner cases, there is a one-to-one mapping between a Wikipedia page and a DBpedia resource based on the article title. For example, for the Wikipedia article on *Berlin*[4], DBpedia will produce the URI *dbr:Berlin*.
- *`http://dbpedia.org/property/`* (prefix *dbp*) for representing properties extracted from the raw infobox extraction (cf. Section 2.3), e.g. *dbp:population*.
- *`http://dbpedia.org/ontology/`* (prefix *dbo*) for representing the DBpedia ontology (cf. Section 2.4), e.g. *dbo: populationTotal*.

[4] `http://en.wikipedia.org/wiki/Berlin`

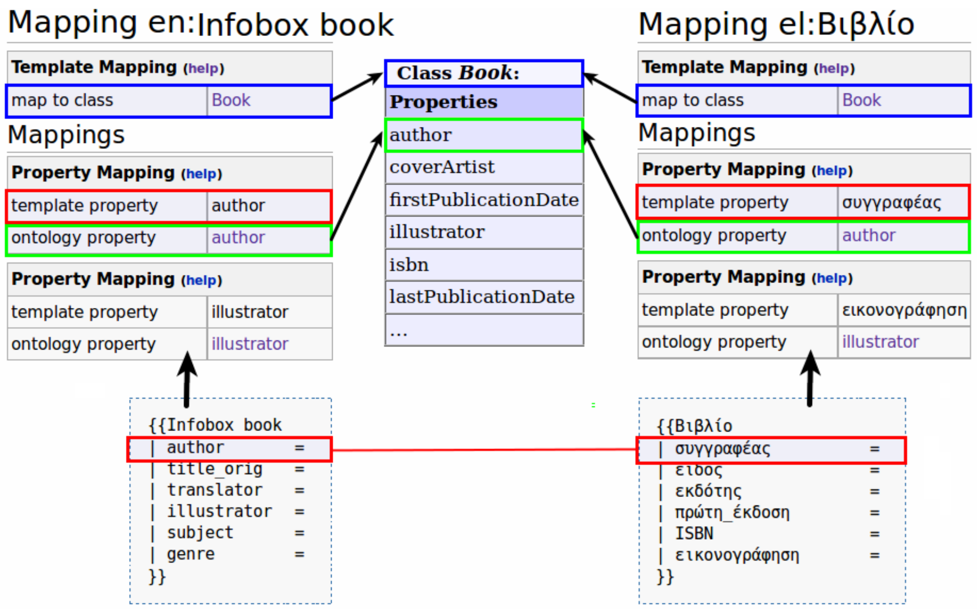


Fig. 2. Depiction of the mapping from the Greek and English Wikipedia templates about books to the same DBpedia Ontology class [23].

Although data from other Wikipedia language editions were extracted, they used the same namespaces. This was achieved by exploiting the Wikipedia *inter-language links*[5]. For every page in a language other than English, the page was extracted only if the page contained an inter-language link to an English page. In that case, using the English link, the data was extracted under the English resource name (i.e. *dbr:Berlin*).

Recent DBpedia internationalisation developments showed that this approach resulted in less and redundant data [23]. Thus, starting from the *DBpedia 3.7 release*, two types of data sets were generated. The *localized data sets* contain all things that are described in a specific language. Within the datasets, things are identified with language specific URIs such as *http://<lang>.dbpedia.org/resource/* for article data and *http://<lang>.dbpedia.org/property/* for property data. In addition, we produce a *canonicalized data set* for each language. The canonicalized data sets only contain things for which a corresponding page in the English edition of Wikipedia exists. Within all canonicalized data sets, the same thing is identified with the same URI from the generic language-agnostic namespace http://dbpedia.org/resource/.

2.6. NLP Extraction

DBpedia provides a number of data sets which have been created to support Natural Language Processing (NLP) tasks [30]. Currently, four datasets are extracted: *topic signatures*, *grammatical gender*, *lexicalizations* and *thematic concept*. While the topic signatures and the grammatical gender extractors primarily extract data from the article text, the lexicalizations and thematic concept extractors make use of the wiki markup.

DBpedia entities can be referred to using many different names and abbreviations. The Lexicalization data set provides access to alternative names for entities and concepts, associated with several scores estimating the association strength between name and URI. These scores distinguish more common names for specific entities from rarely used ones and also show how ambiguous a name is with respect to all possible concepts that it can mean.

The topic signatures data set enables the description of DBpedia resources based on unstructured information, as compared to the structured factual data provided by the mapping-based and raw extractors. We build a Vector Space Model (VSM) where each DBpedia resource is a point in a multidimensional space of words. Each DBpedia resource is represented by a vector, and each word occurring in Wikipedia is a dimension of this vector. Word scores are computed using the

[5] http://en.wikipedia.org/wiki/Help:Interlanguage_links

tf-idf weight, with the intention to measure how strong is the association between a word and a DBpedia resource. Note that word stems are used in this context in order to generalize over inflected words. We use the computed weights to select the strongest related word stems for each entity and build topic signatures [27].

There are two more Feature Extractors related to Natural Language Processing. The thematic concepts data set relies on Wikipedia's category system to capture the idea of a 'theme', a subject that is discussed in its articles. Many of the categories in Wikipedia are linked to an article that describes the main topic of that category. We rely on this information to mark DBpedia entities and concepts that are 'thematic', that is, they are the center of discussion for a category.

The grammatical gender data set uses a simple heuristic to decide on a grammatical gender for instances of the class Person in DBpedia. While parsing an article in the English Wikipedia, if there is a mapping from an infobox in this article to the class `dbo:Person`, we record the frequency of gender-specific pronouns in their declined forms (Subject, Object, Possessive Adjective, Possessive Pronoun and Reflexive) – i.e. he, him, his, himself (masculine) and she, her, hers, herself (feminine). Grammatical genders for DBpedia entities are assigned based on the dominating gender in these pronouns.

2.7. Summary of Other Recent Developments

In this section we summarize the improvements of the DBpedia extraction framework since the publication of the previous DBpedia overview article [5] in 2009. One of the major changes on the implementation level is that the extraction framework has been rewritten in Scala in 2010 to improve the efficiency of the extractors by an order of magnitude compared to the previous PHP based framework. The new more modular framework also allows to extract data from tables in Wikipedia pages and supports extraction from multiple MediaWiki templates per page. Another significant change was the creation and utilization of the DBpedia Mappings Wiki as described earlier. Further significant changes include the mentioned NLP extractors and the introduction of URI schemes.

In addition, there were several smaller improvements and general maintenance: Overall, over the past four years, the parsing of the MediaWiki markup improved quite a lot which led to better overall coverage, for example, concerning references and parser functions. In addition, the collection of MediaWiki namespace identifiers for many languages is now performed semi-automatically leading to a high accuracy of detection. This concerns common title prefixes such as *User, File, Template, Help, Portal* etc. in English that indicate pages that do not contain encyclopedic content and would produce noise in the data. They are important for specific extractors as well, for instance, the category hierarchy data set (SKOS) is produced from pages of the *Category* namespace. Furthermore, the output of the extraction system now supports more formats and several compliance issues regarding URIs, IRIs, N-Triples and Turtle were fixed.

The individual data extractors have been improved as well in both number and quality in many areas. The abstract extraction was enhanced producing more accurate plain text representations of the beginning of Wikipedia article texts. More diverse and more specific datatypes do exist (e.g. many currencies and XSD datatypes such as `xsd:gYearMonth`, `xsd:positiveInteger`, etc.) and for a number of classes and properties, specific datatypes were added (e.g. $\mathrm{inhabitants}/\mathrm{km}^2$ for the population density of populated places and m^3/s for the discharge of rivers). Many issues related to data parsers were resolved and the quality of the `owl:sameAs` data set for multiple language versions was increased by an implementation that takes bijective relations into account.

There are also further extractors, e.g. for Wikipedia page IDs and revisions. Moreover, redirect and disambiguation extractors were introduced and improved. For the redirect data, the transitive closure is computed while taking care of catching cycles in the links. The redirects also help regarding infobox coverage in the mapping-based extraction by resolving alternative template names. Moreover, in the past, if an infobox value pointed to a redirect, this redirection was not properly resolved and thus resulted in RDF links that led to URIs which did not contain any further information. Resolving redirects affected approximately 15% of all links, and hence increased the overall interconnectivity of resources in the DBpedia ontology.

Finally, a new heuristic to increase the connectiveness of DBpedia instances was introduced. If an infobox contains a string value that is not linked to another Wikipedia article, the extraction framework searches for hyperlinks in the same Wikipedia article that have the same anchor text as the infobox value string. If such a link exists, the target of that link is used to replace the string value in the infobox. This method further increases the number of object property assertions in the DBpedia ontology.

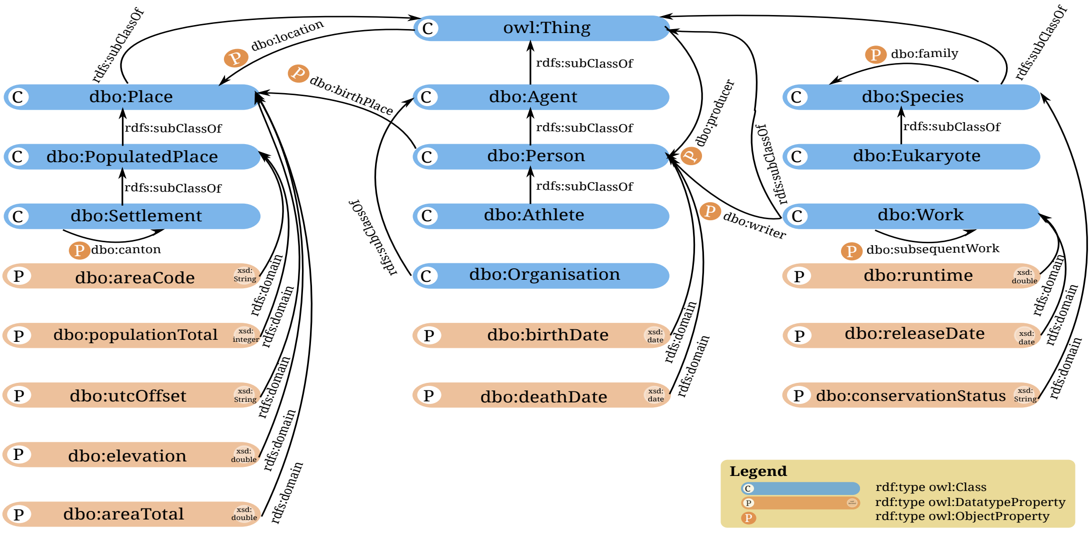


Fig. 3. Snapshot of a part of the DBpedia ontology.

3. DBpedia Ontology

The DBpedia ontology consists of 320 classes which form a subsumption hierarchy and are described by 1,650 different properties. With a maximal depth of 5, the subsumption hierarchy is intentionally kept rather shallow which fits use cases in which the ontology is visualized or navigated. Figure 3 depicts a part of the DBpedia ontology, indicating the relations among the top ten classes of the DBpedia ontology, i.e. the classes with the highest number of instances.

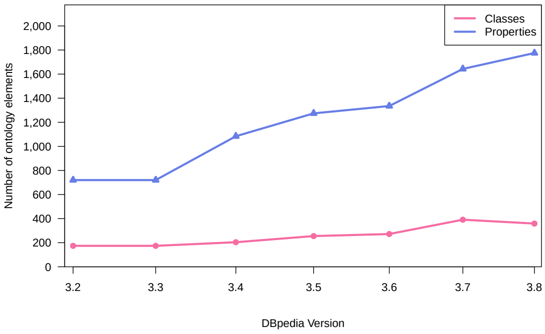


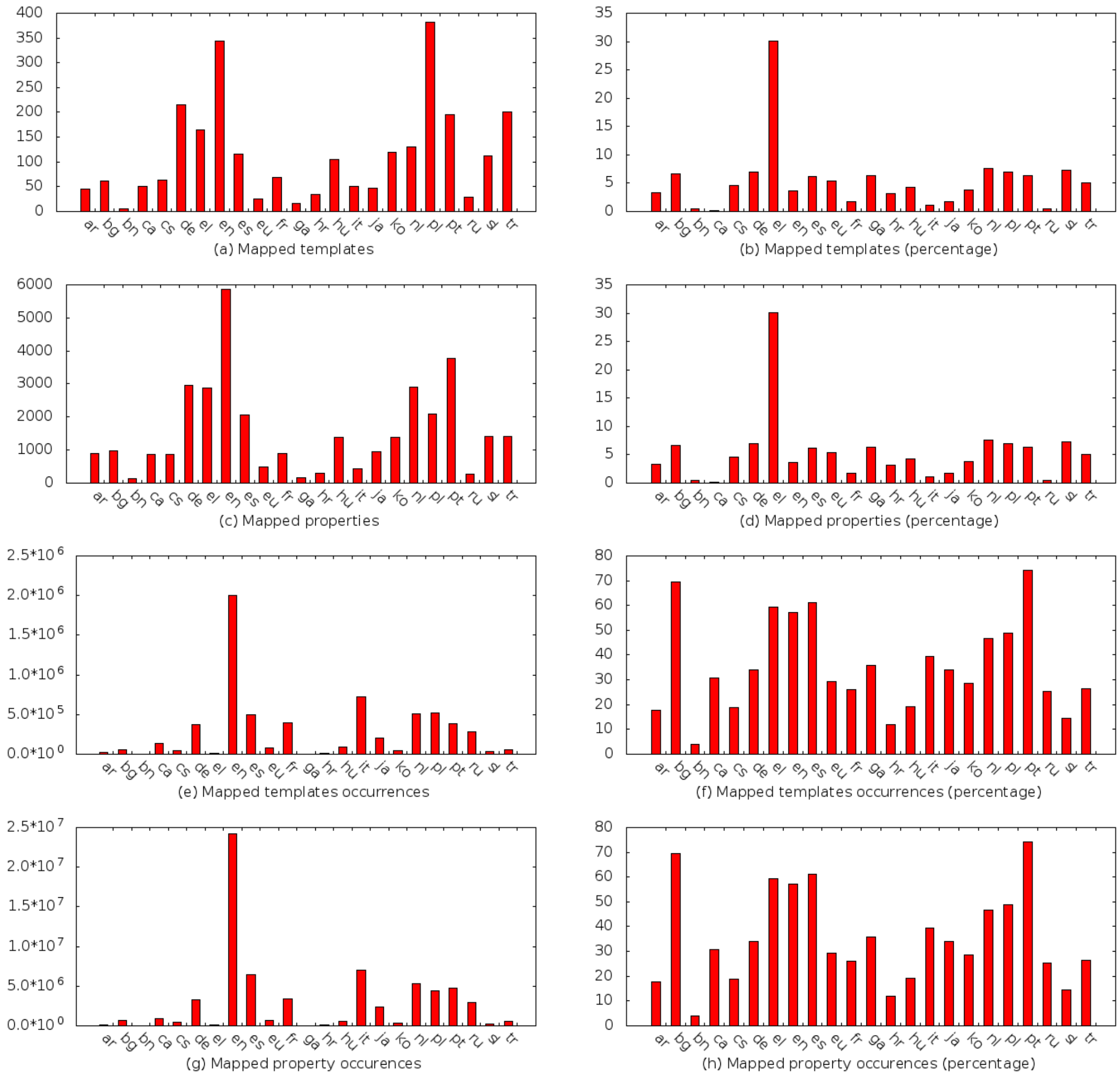
Fig. 4. Growth of the DBpedia ontology.

The DBpedia ontology is maintained and extended by the community in the DBpedia Mappings Wiki. Figure 4 depicts the growth of the DBpedia ontology over time. While the number of classes is not growing too much due to the already good coverage of the initial version of the ontology, the number of properties increases over time due to the collaboration on the DBpedia Mappings Wiki and the addition of more detailed information to infoboxes by Wikipedia editors.

3.1. Mapping Statistics

As of April 2013, there exist mapping communities for 27 languages, 23 of which are active. Figure 5 shows statistics for the coverage of these mappings in DBpedia. Figures (a) and (c) refer to the absolute number of template and property mappings that are defined for every DBpedia language edition. Figures (b) and (d) depict the percentage of the defined template and property mappings compared to the total number of available templates and properties for every Wikipedia language edition. Figures (e) and (g) show the occurrences (instances) that the defined template and property mappings have in Wikipedia. Finally, figures (f) and (h) give the percentage of the mapped templates and properties occurences, compared to the total templates and property occurences in a Wikipedia language edition.

It can be observed in the figure that the Portuguese DBpedia language edition is the most complete regarding mapping coverage. Other language editions such as Bulgarian, Dutch, English, Greek, Polish and Spanish have mapped templates covering more than 50% of total template occurrences. In addition, almost all languages have covered more than 20% of property occurrences, with Bulgarian and Portuguese reaching up to 70%.



(a) Mapped templates

(b) Mapped templates (percentage)

(c) Mapped properties

(d) Mapped properties (percentage)

(e) Mapped templates occurrences

(f) Mapped templates occurrences (percentage)

(g) Mapped property occurences

(h) Mapped property occurences (percentage)

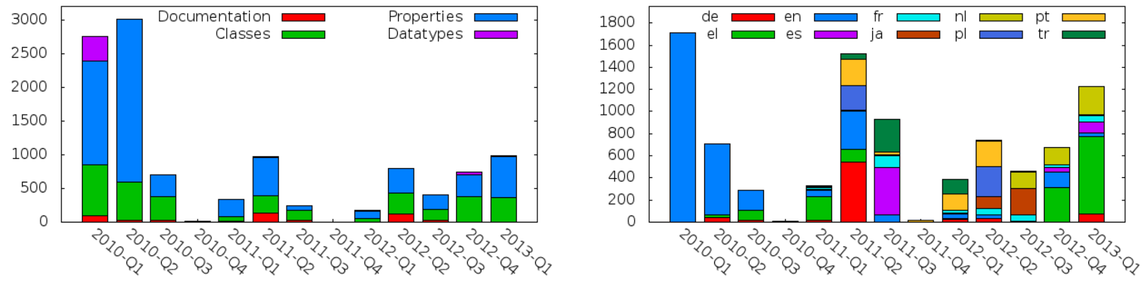
Fig. 5. Mapping coverage statistics for all mapping-enabled languages.

The mapping activity of the ontology enrichment process along with the editing of the ten most active mapping language communities is depicted in Figure 6. It is interesting to notice that the high mapping activity peaks coincide with the DBpedia release dates. For instance, the DBpedia 3.7 version was released on September 2011 and the 2nd and 3rd quarter of that year have a very high activity compared to the 4th quarter. In the last two years (2012 and 2013), most of the DBpedia mapping language communities have defined their own chapters and have their own release dates. Thus, recent mapping activity shows less fluctuation.

Finally, Figure 7 shows the English property mappings occurrence frequency. Both axes are in log scale and represent the number of property mappings (x axis) that have exactly y occurrences (y axis). The occurrence frequency follows a *long tail* distribution. Thus, a low number of property mappings have a high number of occurrences and a high number of property mappings have a low number of occurences.

3.2. Instance Data

The DBpedia 3.8 release contains localized versions of DBpedia for 111 languages which have been extracted from the Wikipedia edition in the correspond-



(a) Ontology edit activity

(b) Mapping edit activity for top 10 most active languages

Fig. 6. Mapping community activity for (a) ontology and (b) 10 most active language editions

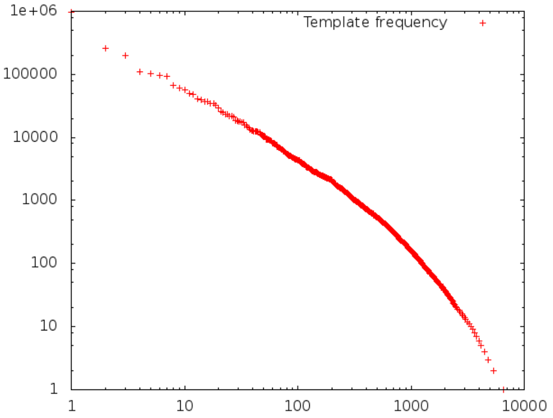


Fig. 7. English property mappings occurrence frequency (both axes are in log scale)

ing language. For 20 of these languages, we report in this section the overall number of entities being described by the localized versions as well as the number of facts (i.e. statements) that have been extracted from infoboxes describing these things. Afterwards, we report on the number of instances of popular classes within the 20 DBpedia versions as well as the conceptual overlap between the languages.

Table 2 shows the overall number of things, ontology and raw-infobox properties, infobox statements and type statements for the 20 languages. The column headings have the following meaning: LD = Localized data sets (see Section 2.5); CD = Canonicalized data sets (see Section 2.5); all = Overall number of instances in the data set, including instances without infobox data; with MD = Number of instances for which mapping-based infobox data exists; Raw Properties = Number of different properties that are generated by the raw infobox extractor; Mapping Properties = Number of different properties that are generated by the mapping-based infobox extractor; Raw Statements = Number of statements (facts) that are generated by the raw infobox extractor; Mapping Statements = Number of statements (facts) that are generated by the mapping-based infobox extractor.

It is interesting to see that the English version of DBpedia describes about three times more instances than the second and third largest language editions (French, German). Comparing the first column of the table with the second and third reveals which portion of the instances of a specific language correspond to instances in the English version of DBpedia and which portion of the instances is described by clean, mapping-based infobox data. The difference between the number of properties in the raw infobox data set and the cleaner mapping-based infobox data set (columns 4 and 5) results on the one hand from multiple Wikipedia infobox properties being mapped to a single ontology property. On the other hand, it reflects the number of mappings that have been so far created in the Mapping Wiki for a specific language.

Table 3 reports the number of instances for a set of selected ontology classes within the canonicalized DBpedia data sets for each language. The indented classes are subclasses of the superclasses set in bold. The zero values in the table indicate that no infoboxes have been mapped to a specific ontology class within the corresponding language so far. Again, the English version of DBpedia covers by far the most instances.

Table 4 shows, for the canonicalized, mapping-based data set, how many instances are described in multiple languages. The Instances column contains the total number of instances per class across all 20 languages, the second column contains the number of instances that are described only in a single language version, the next column contains the number of instances that are contained in two languages but not in three or more languages, etc. For example, 12,936 persons are described in five languages but not in six

	Inst. LD all	*Inst. CD all*	*Inst. with MD CD*	*Raw Prop. CD*	*Map. Prop. CD*	*Raw Statem. CD*	*Map. Statem. CD*
en	3,769,926	3,769,926	2,359,521	48,293	1,313	65,143,840	33,742,015
de	1,243,771	650,037	204,335	9,593	261	7,603,562	2,880,381
fr	1,197,334	740,044	214,953	13,551	228	8,854,322	2,901,809
it	882,127	580,620	383,643	9,716	181	12,227,870	4,804,731
es	879,091	542,524	310,348	14,643	476	7,740,458	4,383,206
pl	848,298	538,641	344,875	7,306	266	7,696,193	4,511,794
ru	822,681	439,605	123,011	13,522	76	6,973,305	1,389,473
pt	699,446	460,258	272,660	12,851	602	6,255,151	4,005,527
ca	367,362	241,534	112,934	8,696	183	3,689,870	1,301,868
cs	225,133	148,819	34,893	5,564	334	1,857,230	474,459
hu	209,180	138,998	63,441	6,821	295	2,506,399	601,037
ko	196,132	124,591	30,962	7,095	419	1,035,606	417,605
tr	187,850	106,644	40,438	7,512	440	1,350,679	556,943
ar	165,722	103,059	16,236	7,898	268	635,058	168,686
eu	132,877	108,713	41,401	2,245	19	2,255,897	532,709
sl	129,834	73,099	22,036	4,235	470	1,213,801	222,447
bg	125,762	87,679	38,825	3,984	274	774,443	488,678
hr	109,890	71,469	10,343	3,334	158	701,182	151,196
el	71,936	48,260	10,813	2,866	288	206,460	113,838

Table 2

Basic statistics about Localized DBpedia Editions.

	en	*it*	*pl*	*es*	*pt*	*fr*	*de*	*ru*	*ca*	*hu*
Person	763,643	145,060	70,708	65,337	43,057	62,942	33,122	18,620	7,107	15,529
Athlete	185,126	47,187	30,332	19,482	14,130	21,646	31,237	0	721	4,527
Artist	61,073	12,511	16,120	25,992	10,571	13,465	0	0	2,004	3,821
Politician	23,096	0	8,943	5,513	3,342	0	0	12,004	1,376	760
Place	572,728	14,1101	182,727	132,961	116,660	80,602	131,766	67,932	73,078	18,324
Popul.Place	387,166	138,077	167,034	121,204	109,418	72,252	79,410	63,826	72,743	15,535
Building	60,514	1,270	2,946	3,570	803	921	83	43	0	527
River	24,267	0	1,383	2	4,149	3,333	6,707	3,924	0	565
Organisation	192,832	4,142	12,193	11,710	10,949	17,513	16,973	1,598	1,399	3,993
Company	44,516	4,142	2,566	975	1,903	5,832	7,200	0	440	618
Educ.Inst.	42,270	0	599	1,207	270	1,636	2,171	1,010	0	115
Band	27,061	0	2,993	0	4,476	3,868	5,368	0	263	802
Work	333,269	51,918	32,386	36,484	33,869	39,195	18,195	34,363	5,240	9,777
Music.Work	159,070	23,652	14,987	15,911	17,053	16,206	0	6,588	697	4,704
Film	71,715	17,210	9,467	9,396	8,896	9,741	15,038	12,045	3,859	2,320
Software	27,947	5,682	3,050	4,833	3,419	5,733	2,368	0	606	857

Table 3

Number of instances per class within 10 localized DBpedia versions.

or more languages. The number 871,630 for the class `Person` means that all 20 language version together describe 871,630 different persons. The number is higher than the number of persons described in the canonicalized English infobox data set (763,643) listed in Table 3, since there are infoboxes in non-English articles describing a person without a corresponding infobox in the English article describing the same person. Summing up columns 2 to 10+ for the `Person` class, we see that 195,263 persons are described in two or more languages. The large difference of this number compared to the total number of 871,630 persons is due to the much smaller size of the localized DBpedia versions compared to the English one (cf. Table 2).

Class	*Instances*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10+*
Person	871.630	676.367	94.339	42.382	21.647	12.936	8.198	5.295	3.437	2.391	4.638
Place	643.260	307.729	150.349	45.836	36.339	20.831	13.523	20.808	31.422	11.262	5.161
Organisation	206.670	160.398	22.661	9.312	5.002	3.221	2.072	1.421	928	594	1.061
Work	360.808	243.706	54.855	23.097	12.605	8.277	5.732	4.007	2.911	1.995	3.623

Table 4

Cross-language overlap: Number of instances that are described in multiple languages.

3.3. Internationalisation Community

The introduction of the mapping-based infobox extractor alongside live synchronisation approaches in [19] allowed the international DBpedia community to easily define infobox-to-ontology mappings. As a result of this development, there are presently mappings for 27 languages[6]. The DBpedia 3.7 release[7] in September 2011 was the first DBpedia release to use the localized I18n DBpedia extraction framework [23].

At the time of writing, official DBpedia chapters for 14 languages have been founded: Basque, Czech, Dutch, English, French, German, Greek, Italian, Japanese, Korean, Polish, Portuguese, Russian and Spanish.[8] Besides providing mappings from infoboxes in the corresponding Wikipedia editions, DBpedia chapters organise a local community and provide hosting for data sets and associated services. Recently, the DBpedia internationalisation committee[9] has manifested its structure and each language edition has a representative with a vote in elections. In some cases (e.g. Greek[10] and Dutch[11]) the existence of a local DBpedia chapter has had a positive effect on the creation of localized LOD clouds [23].

In the weeks leading to a new release, the DBpedia project organises a *mapping sprint*, where communities from each language work together to improve mappings, increase coverage and detect bugs in the extraction process. The progress of the mapping effort is tracked through statistics on the number of mapped templates and properties, as well as the number of times these templates and properties occur in Wikipedia. These statistics provide an estimate of the coverage of each Wikipedia edition in terms of how many entities will be typed and how many properties from those entities will be extracted. Therefore, they can be used by each language edition to prioritize properties and templates with higher impact on the coverage. The mapping statistics have also been used as a way to promote a healthy competition between language editions. A sprint page was created with bar charts that show how close each language is from achieving total coverage (cf. Figure 5), and line charts showing the progress over time highlighting when one language is overtaking another in their race for higher coverage. The mapping sprints have served as a great motivator for the crowd-sourcing efforts, as it can be noted from the increase in the number of mapping contributions in the weeks leading to a release.

4. Live Synchronisation

Wikipedia articles are continuously revised at a very high rate, e.g. the English Wikipedia, in June 2013, has approximately 3.3 million edits per month which is equal to 77 edits per minute[12]. This high change frequency leads to DBpedia data quickly being outdated, which in turn leads to the need for a methodology to keep DBpedia in synchronisation with Wikipedia. As a consequence, the *DBpedia Live* system was developed, which works on a continuous stream of updates from Wikipedia and processes that stream on the fly [19,33]. It allows extracted data to stay up-to-date with a small delay of at most a few minutes. Since the English Wikipedia is the largest among all Wikipedia editions with respect to the number of articles and the number of edits per month, it was the first language DBpedia Live supported[13]. Meanwhile, DBpedia Live for Dutch[14] was developed.

[6] ar, bg, bn, ca, cs, de, el, en, es, et, eu, fr, ga, hi, hr, hu, id, it, ja, ko, nl, pl, pt, ru, sl, tr, ur

[7] `http://blog.dbpedia.org/2011/09/11/`

[8] Accessed on 25/02/2013: `http://wiki.dbpedia.org/Internationalization/Chapters`

[9] `http://wiki.dbpedia.org/Internationalization`

[10] `http://el.dbpedia.org`

[11] `http://nl.dbpedia.org`

[12] `http://stats.wikimedia.org/EN/SummaryEN.htm`

[13] `http://live.dbpedia.org`

[14] `http://live.nl.dbpedia.org`

4.1. DBpedia Live System Architecture

In order for live synchronisation to be possible, we need access to the changes made in Wikipedia. The Wikimedia foundation kindly provided us access to their update stream using the *OAI-PMH* protocol [24]. This protocol allows a programme to pull page updates in XML via HTTP. A Java component, serving as a proxy, constantly retrieves new updates and feeds them to the DBpedia Live framework. This proxy is necessary to decouple the stream from the framework to simplify maintenance of the software. The live extraction workflow uses this update stream to extract new knowledge upon relevant changes in Wikipedia articles.

The overall architecture of DBpedia Live is indicated in Figure 8. The major components of the system are as follows:

- **Local Wikipedia Mirror**: A local copy of a Wikipedia language edition is installed which is kept in real-time synchronisation with its live version using the OAI-PMH protocol. Keeping a local Wikipedia mirror allows us to exceed any access limits posed by Wikipedia.
- **Mappings Wiki**: The DBpedia Mappings Wiki, described in Section 2.4, serves as secondary input source. Changes of the mappings wiki are also consumed via an OAI-PMH stream. Note that a single mapping change can affect a high number of DBpedia resources.
- **DBpedia Live Extraction Manager**: This is the core component of the DBpedia Live extraction architecture. The manager takes feeds of pages for re-processing as input and applies all the enabled extractors. After processing a page, the extracted triples are a) inserted into a backend triple store (in our case Virtuoso [10]), updating the old triples and b) saved as changesets into a compressed N-Triples file structure.
- **Synchronisation Tool**: This tool allows third parties to keep DBpedia Live mirrors up-to-date by harvesting the produced changesets.

4.2. Features of DBpedia Live

The core components of the DBpedia Live Extraction framework provide the following features:

- *Mapping-Affected Pages*: The update of all pages that are affected by a mapping change.
- *Unmodified Pages*: The update of unmodified pages at regular intervals.
- *Changesets Publication*: The publication of triple-changesets.
- *Synchronisation Tool*: A synchronisation tool for harvesting updates to DBpedia Live mirrors.
- *Data Isolation*: Separate data from different sources.

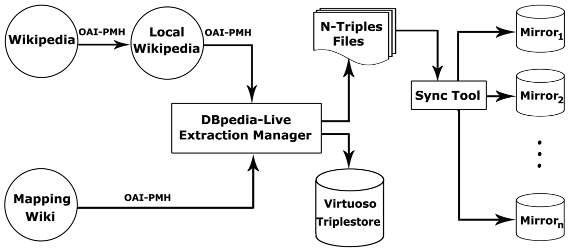


Fig. 8. Overview of DBpedia Live extraction framework.

Mapping-Affected Pages: Whenever an infobox mapping change occurs, all the Wikipedia pages that use that infobox are reprocessed. Taking Figure 2 as an example, if a new property mapping is introduced (i.e. *dbo:translator*) or an existing (i.e. *dbo:illustrator*) is updated or deleted, then all entities belonging to the class *dbo:Book* are reprocessed. Thus, upon a mapping change, we identify all the affected Wikipedia pages and feed them for reprocessing.

Unmodified Pages: Extraction framework improvements or activation / deactivation of DBpedia extractors might never be applied to rarely modified pages. To overcome this problem, we obtain a list of the pages which have not been processed over a period of time (30 days in our case) and feed that list to the DBpedia Live extraction framework for reprocessing. This feed has a lower priority than the update or the mapping affected pages feed and ensures that all articles reflect a recent state of the output of the extraction framework.

Publication of Changesets: Whenever a Wikipedia article is processed, we get two disjoint sets of triples. A set for the added triples, and another set for the deleted triples. We write those two sets into N-Triples files, compress them, and publish the compressed files as changesets. If another DBpedia Live mirror wants to synchronise with the DBpedia Live endpoint, it can just download those files, decompress and integrate them.

Synchronisation Tool: The synchronisation tool enables a DBpedia Live mirror to stay in synchronisation with our live endpoint. It downloads the changeset files sequentially, decompresses them and updates the target SPARQL endpoint via insert and delete operations.

Data Isolation: In order to keep the data isolated, DBpedia Live keeps different sources of data in different SPARQL graphs. Data from the article update feeds are contained in the graph with the URI `http://live.dbpedia.org`, static data (i.e. links to the LOD cloud) are kept in `http://static.dbpedia.org` and the DBpedia ontology is stored in `http://dbpedia.org/ontology`. All data is also accessible under the `http://dbpedia.org` graph for combined queries. Next versions of DBpedia Live will also separate data from the raw infobox extraction and mapping-based infobox extraction.

5. Interlinking

DBpedia is interlinked with numerous external data sets following the Linked Data principles. In this section, we give an overview of the number and types of outgoing links that point from DBpedia into other data sets, as well as the external data sets that set links pointing at DBpedia resources.

5.1. Outgoing Links

Similar to the DBpedia ontology, DBpedia also follows a community approach for adding links to other third party data sets. The DBpedia project maintains a link repository[15] for which conventions for adding linksets and linkset metadata are defined. The adherence to those guidelines is supervised by a linking committee. Linksets, which are added to the repository are used for the subsequent official DBpedia release as well as for DBpedia Live. Table 5 lists the linksets created by the DBpedia community as of April 2013. The first column names the data set that is the target of the links. The second and third column contain the predicate that is used for linking as well as the overall number of links that is set between DBpedia and the external data set. The last column names the tool that was used to generate the links. The value S refers to Silk, L to LIMES, C to custom script and a missing entry means that the dataset is copied from the previous releases and not regenerated.

Links in DBpedia have been used for various purposes. One example is the combination of data about European Union project funding (FTS) [29] and data about countries in DBpedia. The following query compares funding per year (from FTS) and country with the gross domestic product of a country (from DBpedia).

Data set	*Predicate*	*Count*	*Tool*
Amsterdam Museum	owl:sameAs	627	S
BBC Wildlife Finder	owl:sameAs	444	S
Book Mashup	rdf:type owl:sameAs	9 100	
Bricklink	dc:publisher	10 100	
CORDIS	owl:sameAs	314	S
Dailymed	owl:sameAs	894	S
DBLP Bibliography	owl:sameAs	196	S
DBTune	owl:sameAs	838	S
Diseasome	owl:sameAs	2 300	S
Drugbank	owl:sameAs	4 800	S
EUNIS	owl:sameAs	3 100	S
Eurostat (Linked Stats)	owl:sameAs	253	S
Eurostat (WBSG)	owl:sameAs	137	
CIA World Factbook	owl:sameAs	545	S
flickr wrappr	dbp:hasPhotoCollection	3 800 000	C
Freebase	owl:sameAs	3 600 000	C
GADM	owl:sameAs	1 900	
GeoNames	owl:sameAs	86 500	S
GeoSpecies	owl:sameAs	16 000	S
GHO	owl:sameAs	196	L
Project Gutenberg	owl:sameAs	2 500	S
Italian Public Schools	owl:sameAs	5 800	S
LinkedGeoData	owl:sameAs	103 600	S
LinkedMDB	owl:sameAs	13 800	S
MusicBrainz	owl:sameAs	23 000	
New York Times	owl:sameAs	9 700	
OpenCyc	owl:sameAs	27 100	C
OpenEI (Open Energy)	owl:sameAs	678	S
Revyu	owl:sameAs	6	
Sider	owl:sameAs	2 000	S
TCMGeneDIT	owl:sameAs	904	
UMBEL	rdf:type	896 400	
US Census	owl:sameAs	12 600	
WikiCompany	owl:sameAs	8 300	
WordNet	dbp:wordnet_type	467 100	
YAGO2	rdf:type	18 100 000	
Sum		27 211 732	

Table 5

Data sets linked from DBpedia

```
SELECT * { {
 SELECT ?ftsyear ?ftscountry (SUM(?amount) AS
      ?funding)   {
```

[15] https://github.com/dbpedia/dbpedia-links

```
    ?com rdf:type fts-o:Commitment .
    ?com fts-o:year ?year .
    ?year rdfs:label ?ftsyear .
    ?com fts-o:benefit ?benefit .
    ?benefit fts-o:detailAmount ?amount .
    ?benefit fts-o:beneficiary ?beneficiary .
    ?beneficiary fts-o:country ?country .
    ?country owl:sameAs ?ftscountry .
} } {
SELECT ?dbpcountry ?gdpyear ?gdpnominal {
    ?dbpcountry rdf:type dbo:Country .
    ?dbpcountry dbp:gdpNominal ?gdpnominal .
    ?dbpcountry dbp:gdpNominalYear ?gdpyear .
} }
FILTER ((?ftsyear = str(?gdpyear)) &&
        (?ftscountry = ?dbpcountry)) }
```

In addition to providing outgoing links on instance-level, DBpedia also sets links on schema-level pointing from the DBpedia ontology to equivialent terms in other schemas. Links to other schemata can be set by the community within the DBpedia Mappings Wiki by using `owl:equivalentClass` in *class templates* and `owl:equivalentProperty` in *datatype or object property templates*, respectively. In particular, in 2011 Google, Microsoft, and Yahoo! announced their collaboration on Schema.org, a collection of vocabularies for marking up content on web pages. The DBpedia 3.8 ontology contains 45 equivalent class and 31 equivalent property links pointing to `http://schema.org` terms.

5.2. *Incoming Links*

DBpedia is being linked to from a variety of data sets. The overall number of links pointing at DBpedia from other data sets is 39,007,478 according to the Data Hub.[16] However, those counts are entered by users and may not always be valid and up-to-date.

Data set	*Link Predicate Count*	*Link Count*
okaboo.com	4	2,407,121
tfri.gov.tw	57	837,459
naplesplus.us	2	212,460
fu-berlin.de	7	145,322
freebase.com	108	138,619
geonames.org	3	125,734
opencyc.org	3	19,648
geospecies.org	10	16,188
dbrec.net	3	12,856
faviki.com	5	12,768

Table 6

Top 10 data sets in Sindice ordered by the number of links to DBpedia.

Metric	*Value*
Total links:	3,960,212
Total distinct data sets:	248
Total distinct predicates:	345

Table 7

Sindice summary statistics for incoming links to DBpedia.

In order to identify actually published and online data sets that link to DBpedia, we used Sindice [36]. The Sindice project crawls RDF resources on the web and indexes those resources. In Sindice, a data set is defined by the second-level domain name of the entity's URI, e.g. all resources available at the domain `fu-berlin.de` are considered to belong to the same data set. A triple is considered to be a link if the data set of subject and object are different. Furthermore, the Sindice data we used for analysis only considers *authoritative* entities: The data set of a subject of a triple must match the domain it was retrieved from, otherwise it is not considered. Sindice computes a graph summary over all resources they store. With the help of the Sindice team, we examined this graph summary to obtain all links pointing to DBpedia. As shown in Table 7, Sindice knows about 248 data sets linking to DBpedia. 70 of those data sets link to DBpedia via `owl:sameAs`, but other link predicates are also very common as evident in this table. In total, Sindice has indexed 4 million links pointing at DBpedia. Table 6 lists the 10 data sets which set most links to DBpedia along with the used link predicate and the number of links.

It should be noted that the data in Sindice is not complete, for instance it does not contain all data sets that are catalogued by the DataHub[17]. However, it crawls for RDFa snippets, converts microformats etc., which are not captured by the DataHub. Despite the inaccuracy, the relative comparison of different datasets can still give us insights. Therefore, we analysed the link structure of all Sindice datasets using the Sindice cluster (see appendix for details). Table 8 shows the datasets with most incoming links. Those are authorities in the network structure of the web of data and DBpedia is currently ranked second in terms of incoming links.

[16] See `http://wiki.dbpedia.org/Interlinking` for details.

[17] `http://datahub.io/`

domain	datasets	links
purl.org	498	6,717,520
dbpedia.org	248	3,960,212
creativecommons.org	2,483	3,030,910
identi.ca	1,021	2,359,276
l3s.de	34	1,261,487
rkbexplorer.com	24	1,212,416
nytimes.com	27	1,174,941
w3.org	405	658,535
geospecies.org	13	523,709
livejournal.com	14,881	366,025

Table 8

Top 10 datasets by incoming links in Sindice.

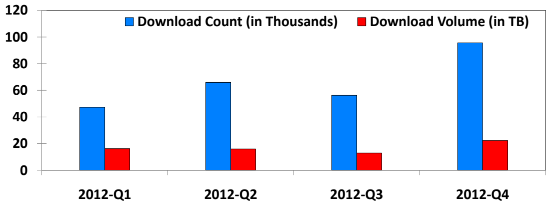


Fig. 9. The download count and download volume (in GB) of the English language of DBpedia.

6. DBpedia Usage Statistics

DBpedia is served on the web in three forms: First, it is provided in the form of downloadable data sets where each data set contains the results of one of the extractors listed in Table 1. Second, DBpedia is served via a public SPARQL endpoint and, third, it provides dereferencable URIs according to the Linked Data principles. In this section, we explore some of the statistics gathered during the hosting of DBpedia over the last couple of years.

6.1. Download Statistics for the DBpedia Data Sets

DBpedia covers more than 100 languages, but those languages vary with respect to the download popularity as well. The top five languages with respect to the download volume are English, Chinese, German, Catalan, and French respectively. The download count and download volume of the English language is indicated in Figure 9. To host the DBpedia dataset downloads, a bandwidth of approximately 18 TB per quarter or 6 TB per month is currently needed.

Furthermore, DBpedia consists of several data sets which vary with respect to their download popularity. The download count and the download volume of each data set during the year 2012 is depicted in Figure 10. In those statistics we filtered out all IP addresses, which requested a file more than 1000 times per month.[18] Pagelinks are the most downloaded dataset, although they are not semantically rich as they do not reveal which type of links exists between two resources. Supposedly, they are used for network analysis or providing relevant links in user interfaces and downloaded more often as they are not provided via the official SPARQL endpoint.

DBpedia	*Configuration*
3.3 - 3.4	AMD Opteron 8220 2.80Ghz, 4 Cores, 32GB
3.5 - 3.7	Intel Xeon E5520 2.27Ghz, 8 Cores, 48GB
3.8	Intel Xeon E5-2630 2.30GHz, 8 Cores, 64GB

Table 9

Hardware of the machines serving the public SPARQL endpoint.

6.2. Public Static DBpedia SPARQL Endpoint

The main public DBpedia SPARQL endpoint[19] is hosted using the Virtuoso Universal Server (Enterprise Edition) version 6.4 software in a 4-nodes cluster configuration. This cluster setup provides parallelization of query execution, even when the cluster nodes are on the same machine, as splitting a query over several nodes allows better use of parallel threads on modern multi-core CPUs on standard commodity hardware.

Virtuoso supports infinite horizontal scale-out, either by redistributing the existing cluster nodes onto multiple machines, or by adding several separate clusters with a round robin HTTP front-end. This allows the cluster setup to grow in line with desired response times for an RDF data set collection of any size. As the size of the DBpedia data set increased and its use by the Linked Data community grew, the project migrated to increasingly powerful hardware as shown in Table 9.

The Virtuoso instance is configured to process queries within a 1,200 second timeout window and a maximum result set size of 50,000 rows. It provides `OFFSET` and `LIMIT` support for paging alongside the ability to produce partial results.

The log files used in the following analysis excluded traffic generated by:

1. clients that have been temporarily rate limited after a burst period,

[18] The IP address was only filtered for that specific file and month in those cases.

[19] `http://dbpedia.org/sparql`

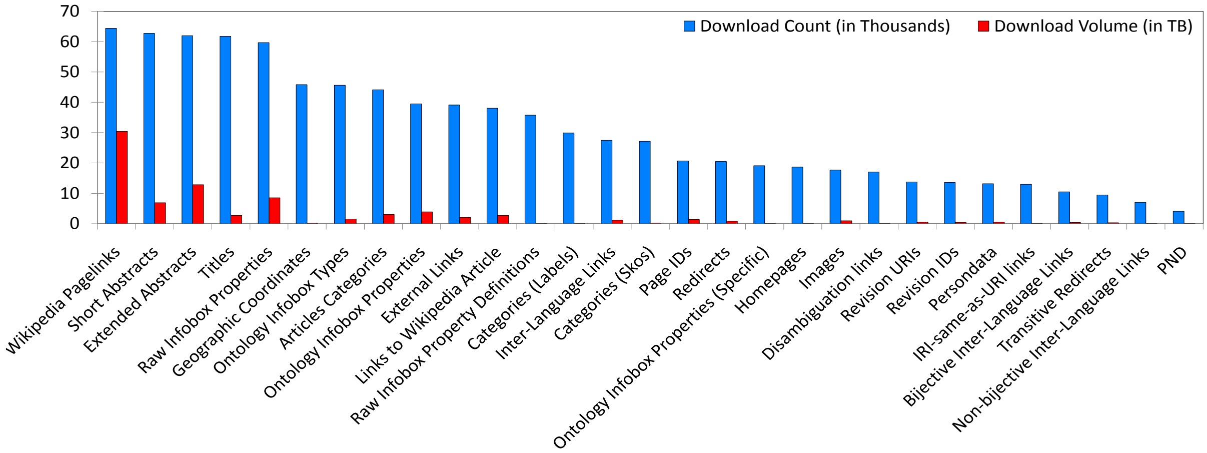


Fig. 10. The download count and download volume (in GB) of the DBpedia data sets.

2. clients that have been banned after misuse,
3. applications, spiders and other crawlers that are blocked after frequently hitting the rate limit or generally use too many resources.

Virtuoso supports HTTP Access Control Lists (ACLs) which allow the administrator to rate limit certain IP addresses or whole IP ranges. A maximum number of requests per second (currently 15) as well as a bandwidth limit per request (currently 10MB) are enforced. If the client software can handle compression, replies are compressed to further save bandwidth. Exception rules can be configured for multiple clients hidden behind a NAT firewall (appearing as a single IP address) or for temporary requests for higher rate limits. When a client hits an ACL limit, the system reports an appropriate HTTP status code[20] like 509 and quickly drops the connection. The system further uses an iptables based firewall for permanent blocking of clients identified by their IP addresses.

6.3. *Public Static Endpoint Statistics*

The statistics presented in this section were extracted from reports generated by Webalizer v2.21[21]. Table 10 and Table 11 show various DBpedia SPARQL endpoint usage statistics for the last couple of DBpedia releases. The *Avg/Day* column represents the average number of hits (resp. visits) per day, followed by the *Median* and *Standard Deviation*. The last column shows the *maximum* number of hits (visits) that was recorded on a single day for each data set version. Visits (i.e. sessions of subsequent queries from the same client) are determined by a floating 30 minute time window. All requests from behind a NAT firewall are logged under the same external IP address and are therefore counted towards the same visit if they occur within the 30 minute interval.

DBpedia	*Avg/Day*	*Median*	*Stdev*	*Maximum*
3.3	733,811	711,306	188,991	1,319,539
3.4	1,212,549	1,165,893	351,226	2,371,657
3.5	1,122,612	1,035,444	386,454	2,908,720
3.6	1,328,355	1,286,750	256,945	2,495,031
3.7	2,085,399	1,930,728	1,057,398	8,873,473
3.8	2,910,410	2,717,775	1,085,640	7,678,490

Table 10

Number of SPARQL endpoint hits.

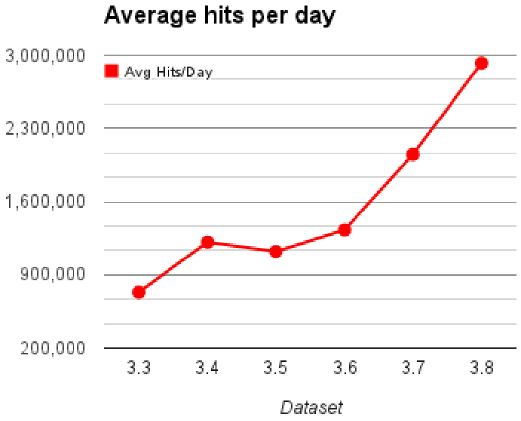


Fig. 11. Average number of SPARQL endpoint requests per day.

[20] http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

[21] http://www.webalizer.org

DBpedia	*Avg/Day*	*Median*	*Stdev*	*Maximum*
3.3	9,750	9,775	2,036	13,126
3.4	11,329	11,473	1,546	14,198
3.5	16,592	16,902	2,969	23,129
3.6	19,471	17,664	5,691	56,064
3.7	23,972	22,262	10,741	127,869
3.8	16,851	16,711	2,960	27,416

Table 11

Number of SPARQL endpoint visits.

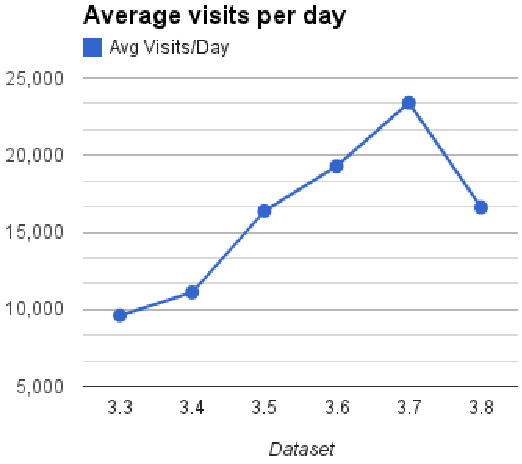


Fig. 12. Average SPARQL endpoint visits per day.

Figure 11 and Figure 12 show the increasing popularity of DBpedia. There is a distinct dip in hits to the SPARQL endpoint in DBpedia 3.5, which is partially due to more strict initial limits for bot-related traffic which were later relaxed. The sudden drop of visits between the 3.7 and the 3.8 data sets can be attributed to:

1. applications starting to use their own private DBpedia endpoint
2. blocking of apps that were abusing the DBpedia endpoint
3. uptake of the language specific DBpedia endpoints and DBpedia Live

6.4. Query Types and Trends

The DBpedia server is not only a SPARQL endpoint, but also serves as a Linked Data Hub returning resources in a number of different formats. For each data set we randomly selected 14 days worth of log files and processed those in order to show the various services called. Table 12 shows the number of hits to the various endpoints.

Endpoint	*3.3*	*3.4*	*3.5*	*3.6*	*3.7*	*3.8*
/data	1,355	2,681	2,230	2,547	3,714	4,246
/ontology	80	168	142	178	116	97
/page	2,634	4,703	1,810	1,687	3,787	7,377
/property	231	311	137	176	176	128
/resource	2,976	4,080	2,554	2,309	4,436	7,143
/sparql	2,090	4,177	2,266	9,112	15,902	15,475
other	252	420	342	277	579	695
total	9,619	16,541	9,434	16,286	28,710	35,142

Table 12

Hits per service to http://dbpedia.org in thousands.

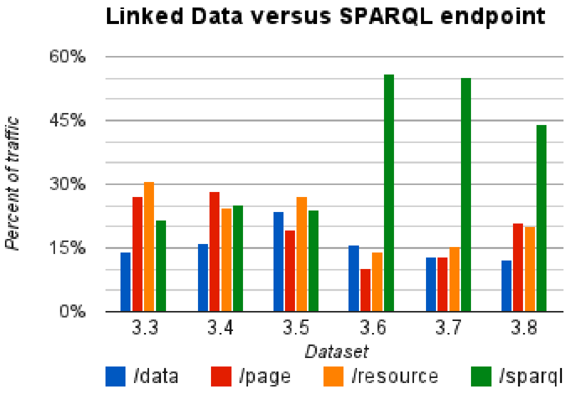


Fig. 13. Traffic Linked Data versus SPARQL endpoint

The /resource endpoint uses the *Accept:* line in the HTTP header sent by the client to return a HTTP status code 30x to redirect the client to either the /page (HTML based) or /data (formats like RDF/XML or Turtle) equivalent of the article. Clients also frequently mint their own URLs to either /page or /data version of an articles directly, or download the raw data from the links at the bottom of the /page based article. This explains why the count of /page and /data hits in the table is larger than the number of hits on the /resource endpoint. The /ontology and /property endpoints return meta information about the DBpedia ontology.

Figure 13 shows the percentages of traffic hits that were generated by the main endpoints. As we can see, the usage of the SPARQL endpoint has doubled from about 22 percent in 2009 to about 44 percent in 2013. However, this still means that 56 percent of traffic hits are directed to the Linked Data service.

In Table 13, we focussed on the calls to the /sparql endpoint and counted the number of statements per type. As the log files only record the full SPARQL query on a GET request, all the PUT requests are counted as unknown.

Statement	*3.3*	*3.4*	*3.5*	*3.6*	*3.7*	*3.8*
ask	56	269	360	326	159	677
construct	55	31	14	11	22	38
describe	11	8	4	7	62	111
select	1891	3663	1658	8030	11204	13516
unknown	78	206	229	738	4455	1134
total	2090	4177	2266	9112	15902	15475

Table 13

Hits per statement type in thousands.

Statement	*3.3*	*3.4*	*3.5*	*3.6*	*3.7*	*3.8*
distinct	19.5	11.4	17.3	19.4	13.3	25.4
filter	45.7	13.7	31.8	25.3	29.2	36.1
functions	8.8	6.9	23.5	21.3	25.5	25.9
geo	27.7	7.0	39.6	6.7	9.3	9.2
group	0.0	0.0	0.0	0.0	0.0	0.2
limit	4.6	6.5	11.6	10.5	7.7	7.8
optional	30.6	23.8	47.3	26.3	16.7	17.2
order	2.3	1.3	1.9	3.2	1.2	1.6
union	3.3	3.2	26.4	11.3	12.1	20.6

Table 14

Trends in SPARQL select (rounded values in %).

Finally, we analyzed each SPARQL query and counted the use of keywords and constructs like:

- DISTINCT
- FILTER
- FUNCTIONS like CONCAT, CONTAINS, ISIRI
- Use of GEO objects
- GROUP BY
- LIMIT / OFFSET
- OPTIONAL
- ORDER BY
- UNION

For the *GEO objects* we counted the use of SPARQL PREFIX *geo:* and *wgs84*:* declarations and usage in property tags. Table 14 shows the use of various keywords as a percentage of the total *select* queries made to the /sparql endpoint for the sample sets. In general, we observed that queries became more complex over time indicating an increasing maturity and higher expectations of the user base.

6.5. *Statistics for DBpedia Live*

Since its official release at the end of June 2011, DBpedia Live attracted a steadily increasing number of users. Furthermore, more users tend to use the synchronisation tool to synchronise their own DBpedia Live mirrors. This leads to an increasing number of live update requests, i.e. changeset downloads. Figure 14 indicates the number of daily SPARQL and synchronisation requests sent to DBpedia Live endpoint in the period between August 2012 and January 2013.

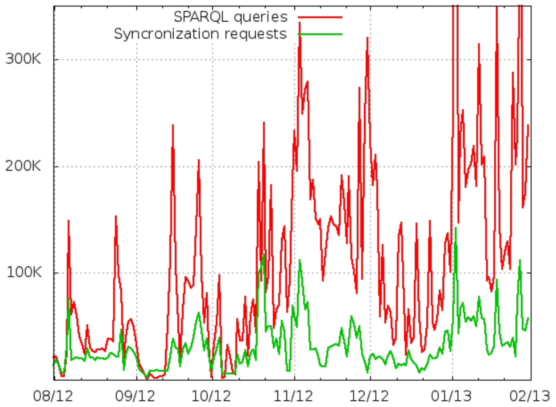


Fig. 14. Number of daily requests sent to the DBpedia Live for a) SPARQL queries and b) syncronization requests from August 2012 until January 2013

7. Use Cases and Applications

With DBpedia evolving into a hub on the Web of Data due to its coverage of various domains, it is an eminent data set for various tasks. It can not only improve Wikipedia search but also serve as a data source for applications and mashups, as well as for text analysis and annotation tools.

7.1. *Natural Language Processing*

DBpedia can support many tasks in Natural Language Processing (NLP) [30]. For that purpose, DBpedia includes a number of specialized data sets[22]. For instance, the lexicalizations data set can be used to estimate the ambiguity of phrases, to help select unambiguous identifiers for ambiguous phrases, or to provide alternative names for entities, just to mention a few examples. Topic signatures can be useful in tasks such as query expansion or document summarization, and has been successfully employed to classify ambiguously described images as good depictions of DBpedia entities [12]. The thematic concepts data set of resources can be used for creating a corpus from

[22] http://wiki.dbpedia.org/Datasets/NLP

Wikipedia to be used as training data for topic classifiers, among other things (see below). The grammatical gender data set can, for example, be used to add a gender feature in co-reference resolution.

7.1.1. Annotation: Entity Disambiguation

An important use case for NLP is annotating texts or other content with semantic information. Named entity recognition and disambiguation – also known as key phrase extraction and entity linking tasks – refers to the task of finding real world entities in text and linking them to unique identifiers. One of the main challenges in this regard is ambiguity: an entity name, or surface form, may be used in different contexts to refer to different concepts. Many different methods have been developed to resolve this ambiguity with fairly high accuracy [21].

As DBpedia reflects a vast amount of structured real world knowledge obtained from Wikipedia, DBpedia URIs can be used as identifiers for the majority of domains in text annotation. Consequently, interlinking text documents with Linked Data enables the Web of Data to be used as background knowledge within document-oriented applications such as semantic search or faceted browsing (cf. Section 7.3).

Many applications performing this task of annotating text with entities in fact use DBpedia entities as targets. For example, DBpedia Spotlight [31] is an open source tool[23] including a free web service that detects mentions of DBpedia resources in text. It uses the lexicalizations in conjunction with the topic signatures data set as context model in order to be able to disambiguate found mentions. The main advantage of this system is its comprehensiveness and flexibility, allowing one to configure it based on quality measures such as prominence, contextual ambiguity, topical pertinence and disambiguation confidence, as well as the DBpedia ontology. The resources that should be annotated can be specified by a list of resource types or by more complex relationships within the knowledge base described as SPARQL queries.

There are numerous other NLP APIs that link entities in text to DBpedia: *AlchemyAPI*[24], *Semantic API* from Ontos[25], *Open Calais*[26] and *Zemanta*[27] among others. Furthermore, the DBpedia ontology has been used for training named entity recognition systems (without disambiguation) in the context of the *Apache Stanbol* project[28].

Tag disambiguation Similar to linking entities in text to DBpedia, user-generated tags attached to multimedia content such as music, photos or videos can also be connected to the Linked Data hub. This has previously been implemented by letting the user resolve ambiguities. For example, *Faviki*[29] suggests a set of DBpedia entities coming from Zemanta's API and lets the user choose the desired one. Alternatively, similar disambiguation techniques as mentioned above can be utilized to choose entities from tags automatically [13]. The BBC[30] employs DBpedia URIs for tagging their programmes. Short clips and full episodes are tagged using two different tools while utilizing DBpedia to benefit from global identifiers that can be easily integrated with other knowledge bases.

7.1.2. Question Answering

DBpedia provides a wealth of human knowledge across different domains and languages, which makes it an excellent target for question answering and keyword search approaches. One of the most prominent efforts in this area is the *DeepQA project*, which resulted in the *IBM Watson* system. The Watson system won a $1 million prize in Jeopardy and relies on several data sets including DBpedia[31]. DBpedia is also the primary target for several QA systems in the Question Answering over Linked Data (QALD) workshop series[32]. Several QA systems, such as *TBSL* [41], *PowerAqua* [28], *FREyA* [9] and *QAKiS* [6] have been applied to DBpedia using the QALD benchmark questions. DBpedia is interesting as a test case for such systems. Due to its large schema and data size as well as its topic adversity, it provides significant scientific challenges. In particular, it would be difficult to provide capable QA systems for DBpedia based only on simple patterns or via domain specific dictionaries, because of its size and broad coverage. Therefore, a question answering system, which is able to reliable answer questions over DBpedia correctly could be seen as a truly intelligent system. In the latest QALD series, question answering benchmarks also exploit national DBpedia chapters for multilingual question answering.

[23] `http://spotlight.dbpedia.org/`

[24] `http://www.alchemyapi.com/`

[25] `http://www.ontos.com/`

[26] `http://www.opencalais.com/`

[27] `http://www.zemanta.com/`

[28] `http://stanbol.apache.org/`

[29] `http://www.faviki.com/`

[30] `http://bbc.co.uk`

[31] `http://www.aaai.org/Magazine/Watson/watson.php`

[32] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/`

Similarly, the *slot filling* task in natural language processing poses the challenge of finding values for a given entity and property from mining text. This can be viewed as question answering with static questions but changing targets. DBpedia can be exploited for fact validation or training data in this task, as was done by the Watson team [4] and others [25].

7.2. Digital Libraries and Archives

In the case of libraries and archives, DBpedia could offer a broad range of information on a broad range of domains. In particular, DBpedia could provide:

- Context information for bibliographic and archive records: Background information such as an author's demographics, a film's homepage or an image could be used to enhance user interaction.
- Stable and curated identifiers for linking: DBpedia is a hub of Linked Open Data. Thus, (re-)using commonly used identifiers could ease integration with other libraries or knowledge bases.
- A basis for a thesaurus for subject indexing: The broad range of Wikipedia topics in addition to the stable URIs could form the basis for a global classification system.

Libraries have already invested both in Linked Data and Wikipedia (and transitively to DBpedia) though the realization of the *Virtual International Authority Files* (VIAF) project.[33] Recently, it was announced that VIAF added a total of 250,000 reciprocal authority links to Wikipedia.[34] These links are already harvested by DBpedia Live and will also be included in the next static DBpedia release. This creates a huge opportunity for libraries that use VIAF to get connected to DBpedia and the LOD cloud in general.

7.3. Knowledge Exploration

Since DBpedia spans many domains and has a diverse schema, many knowledge exploration tools either used DBpedia as a testbed or were specifically built for DBpedia. We give a brief overview of tools and structure them in categories:

Facet Based Browsers An award-winning[35] facet-based browser used the Neofonie search engine to combine facts in DBpedia with full-text from Wikipedia in order to compute hierarchical facets [14]. Another facet based browser, which allows to create complex graph structures of facets in a visually appealing interface and filter them is gFacet [15]. A generic SPARQL based facet explorer, which also uses a graph based visualisation of facets, is LODLive [7]. The OpenLink built-in facet based browser[36] is an interface, which enables developers to explore DBpedia, compute aggregations over facets and view the underlying SPARQL queries.

Search and Querying The DBpedia Query Builder[37] allows developers to easily create simple SPARQL queries, more specifically sets of triple patterns via intelligent autocompletion. The autocompletion functionality ensures that only URIs, which lead to solutions are suggested to the user. The *RelFinder* [16] tool provides an intuitive interface, which allows to explore the neighborhood and connections between resources specified by the user. For instance, the user can view the shortest paths connecting certain persons in DBpedia. *SemLens* [17] allows to create statistical analysis queries and correlations in RDF data and DBpedia in particular.

Spatial Applications *DBpedia Mobile* [3] is a location aware client, which renders a map of nearby locations from DBpedia, provides icons for schema classes and supports more than 30 languages from various DBpedia language editions. It can follow RDF links to other data sets linked from DBpedia and supports powerful SPARQL filters to restrict the viewed data.

7.4. Applications of the Extraction Framework: Wiktionary Extraction

Wiktionary is one of the biggest collaboratively created lexical-semantic and linguistic resources available, written in 171 languages (of which approximately 147 can be considered active[38]), containing information about hundreds of spoken and even ancient

[33] http://viaf.org

[34] Accessed on 12/02/2013: http://www.oclc.org/research/news/2012/12-07a.html

[35] http://blog.dbpedia.org/2009/11/20/german-government-proclaims-faceted-wikipedia-search-one-of-the-365-best-ideas-in-germany/

[36] http://dbpedia.org/fct/

[37] http://querybuilder.dbpedia.org/

[38] http://s23.org/wikistats/wiktionaries_html.php

languages. For example, the English Wiktionary contains nearly 3 million words[39]. A Wiktionary page provides, for a lexical word, a hierarchical disambiguation to its language, part of speech, sometimes etymologies and most prominently senses. Within this tree, numerous kinds of linguistic properties are given, including synonyms, hyponyms, hyperonyms, example sentences, links to Wikipedia and many more. The fast changing nature together with the fragmentation of the project into Wiktionary language editions (WLE) with independent layout rules poses the biggest problem to the automated transformation into a structured knowledge base. We identified this as a serious problem: Although the value of Wiktionary is known and usage scenarios are obvious, only some rudimentary tools exist, which either focus on a specific subset of the data or they only cover one or two WLEs [18]. Existing tools can be seen as adapters to single WLE — they are hard to maintain and there are too many languages, that constantly change and require a programmer to refactor complex code. Opposed to the currently employed classic and straight-forward approach (implementing software adapters for scraping), Wiktionary2RDF [18] employs a declarative mediator/wrapper pattern. Figure 15 shows the work flow of the specially created Wiktionary extractor, that takes as input one config XML file for each WLE. The aim is to enable non-programmers (the community of adopters and domain experts) to tailor and maintain the WLE wrappers themselves. A simple XML dialect was created to encode the "entry layout explained" (ELE) guidelines and declare triple patterns, that define how the resulting RDF should be built. An example is given for the following wiki syntax:

```
===Synonyms===
* [[building]]
* [[company]]
```

Regular expression for scraping and the triple generation rule encoded in XML look like:

```
<wikiTemplate>===Synonyms===
(* [[\$target]]
)+
</wikiTemplate>
...
<triple s="http://some.ns/$entityId" p="http://some.ns/
      hasSynonym" o="http://some.ns/$target" />
```

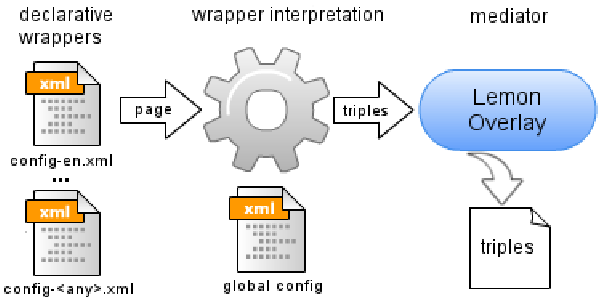


Fig. 15. Detail view of the Wiktionary extractor

This configuration is interpreted and run against Wiktionary dumps. The resulting data set is open in every aspect and hosted as Linked Data.[40] Statistics are shown in Table 15.

8. Related Work

8.1. Cross Domain Community Knowledge Bases

8.1.1. Wikidata

In March 2012, the Wikimedia Germany e.V. started the development of Wikidata[41]. Wikidata is a free knowledge base about the world that can be read and edited by humans and machines alike. It provides data in all languages of the Wikimedia projects, and allows for central access to the data in a similar vein as Wikimedia Commons does for multimedia files. Things described in the Wikidata knowledge base are called items and can have labels, descriptions and aliases in all languages. Wikidata does not aim at offering the truth about things, but providing statements about them. Rather than stating that Berlin has a population of 3.5 million, Wikidata contains the statement about Berlin's population being 3.5 million as of 2011 according to the German statistical office. Thus, Wikidata can offer a variety of statements from different sources and dates. As there are potentially many different statements for a given item and property, ranks can be added to statements to define their status (preferred, normal or deprecated). The initial development was divided in three phases:

- The first phase (interwiki links) created an entity base for the Wikimedia projects. This provides a better alternative to the previous interlanguage link system.

[39] See `http://en.wiktionary.org/wiki/semantic` for a simple example page

[40] `http://wiktionary.dbpedia.org/`

[41] `wikidata.org/`

language	*#words*	*#triples*	*#resources*	*#predicates*	*#senses*	*XML lines*
en	2,142,237	28,593,364	11,804,039	28	424,386	930
fr	4,657,817	35,032,121	20,462,349	22	592,351	490
ru	1,080,156	12,813,437	5,994,560	17	149,859	1449
de	701,739	5,618,508	2,966,867	16	122,362	671

Table 15

Statistical comparison of extractions for different languages. XML lines measures the number of lines of the XML configuration files

– The second phase (infoboxes) gathered infobox-related data for a subset of the entities, with the explicit goal of augmenting the infoboxes that are currently widely used with data from Wikidata.
– The third phase (lists) will expand the set of properties beyond those related to infoboxes, and will provide ways of exploiting this data within and outside the Wikimedia projects.

At the time of writing of this article, the development of the third phase is ongoing.

Wikidata already contains 11.95 million items and 348 properties that can be used to describe them. Since March 2013 the Wikidata extension is live on all Wikipedia language editions and thus their pages can be linked to items in Wikidata and include data from Wikidata.

Wikidata also offers a Linked Data interface[42] as well as regular RDF dumps of all its data. The planned collaboration with Wikidata is outlined in Section 9.

8.1.2. Freebase

Freebase[43] is a graph database, which also extracts structured data from Wikipedia and makes it available in RDF. Both DBpedia and Freebase link to each other and provide identifiers based on those for Wikipedia articles. They both provide dumps of the extracted data, as well as APIs or endpoints to access the data and allow their communities to influence the schema of the data. There are, however, also major differences between both projects. DBpedia focuses on being an RDF representation of Wikipedia and serving as a hub on the Web of Data, whereas Freebase uses several sources to provide broad coverage. The store behind Freebase is the GraphD [32] graph database, which allows to efficiently store metadata for each fact. This graph store is append-only. Deleted triples are marked and the system can easily revert to a previous version. This is necessary, since Freebase data can be directly edited by users, whereas information in DBpedia can only indirectly be edited by modifying the content of Wikipedia or the Mappings Wiki. From an organisational point of view, Freebase is mainly run by Google, whereas DBpedia is an open community project. In particular in focus areas of Google and areas in which Freebase includes other data sources, the Freebase database provides a higher coverage than DBpedia.

8.1.3. YAGO

One of the projects, which pursues similar goals to DBpedia is *YAGO*[44] [39]. YAGO is identical to DBpedia in that each article in Wikipedia becomes an entity in YAGO. Based on this, it uses the leaf categories in the Wikipedia category graph to infer type information about an entity. One of its key features is to link this type information to WordNet. WordNet synsets are represented as classes and the extracted types of entities may become subclasses of such a synset. In the *YAGO2 system* [20], declarative extraction rules were introduced, which can extract facts from different parts of Wikipedia articles, e.g. infoboxes and categories, as well as other sources. YAGO2 also supports spatial and temporal dimensions for facts at the core of its system.

One of the main differences between DBpedia and YAGO in general is that DBpedia tries to stay very close to Wikipedia and provide an RDF version of its content. YAGO focuses on extracting a smaller number of relations compared to DBpedia to achieve very high precision and consistent knowledge. The two knowledge bases offer different type systems: Whereas the DBpedia ontology is manually maintained, YAGO is backed by WordNet and Wikipedia leaf categories. Due to this, YAGO contains much more classes than DBpedia. Another difference is that the integration of attributes and objects in infoboxes is done via mappings in DBpedia and, therefore, by the DBpedia com-

[42] http://meta.wikimedia.org/wiki/Wikidata/Development/LinkedDataInterface

[43] http://www.freebase.com/

[44] http://www.mpi-inf.mpg.de/yago-naga/yago/

munity itself, whereas this task is facilitated by expert-designed declarative rules in YAGO2.

The two knowledge bases are connected, e.g. DBpedia offers the YAGO type hierarchy as an alternative to the DBpedia ontology and `sameAs` links are provided in both directions. While the underlying systems are very different, both projects share similar aims and positively complement and influence each other.

8.2. Knowledge Extraction from Wikipedia

Since its official start in 2001, Wikipedia has always been the target of automatic extraction of information due to its easy availability, open license and encyclopedic knowledge. A large number of parsers, scraper projects and publications exist. In this section, we restrict ourselves to approaches that are either notable, recent or pertinent to DBpedia. MediaWiki.org maintains an up-to-date list of software projects[45], who are able to process wiki syntax, as well as a list of data extraction extensions[46] for MediaWiki.

JWPL (Java Wikipedia Library, [46]) is an open-source, Java-based API that allows to access information provided by the Wikipedia API (redirects, categories, articles and link structure). JWPL contains a MediaWiki Markup parser that can be used to further analyze the contents of a Wikipedia page. Data is also provided as XML dump and is incorporated in the lexical resource UBY[47] for language tools.

Several different approaches to extract knowledge from Wikipedia are presented in [34]. Given features like anchor texts, interlanguage links, category links and redirect pages are utilized e.g. for word-sense disambiguations or synonyms, translations, taxonomic relations and abbreviation or hypernym resolution, respectively. Apart from this, link structures are used to build the *Wikipedia Thesaurus* Web service[48]. Additional projects presented by the authors that exploit the mentioned features are listed on the *Special Interest Group on Wikipedia Mining (SIGWP)* Web site[49].

An earlier approach to improve the quality of the infobox schemata and contents is described in [44]. The presented methodology encompasses a three step process of *preprocessing*, *classification* and *extraction*. During preprocessing refined target infobox schemata are created applying statistical methods and training sets are extracted based on real Wikipedia data. After assigning a class and the corresponding target schema (classification) the training sets are used to extract target infobox values from the document's text applying machine learning algorithms.

The idea of using structured data from certain markup structures was also applied to other user-driven Web encyclopedias. In [35] the authors describe their effort building an integrated *Chinese Linking Open Data (CLOD)* source based on the Chinese Wikipedia and the two widely used and large encyclopedias *Baidu Baike*[50] and *Hudong Baike*[51]. Apart from utilizing MediaWiki and HTML Markup for the actual extraction, the Wikipedia interlanguage links were used to link the CLOD source to the English DBpedia.

A more generic approach to achieve a better cross-lingual knowledge-linkage beyond the use of Wikipedia interlanguage links is presented in [42]. Focusing on wiki knowledge bases the authors introduce their solution based on structural properties like similar linkage structures, the assignment to similar categories and similar interests of the authors of wiki documents in the considered languages. Since this approach is language-feature-agnostic it is not restricted to certain languages.

KnowItAll[52] is a web scale knowledge extraction effort, which is domain-independent, and uses generic extraction rules, co-occurrence statistics and Naive Bayes classification [11]. Cyc [26] is a large common sense knowledge base, which is now partially released as OpenCyc and also available as an OWL ontology. OpenCyc is linked to DBpedia, which provides an ontological embedding in its comprehensive structures. WikiTaxonomy [37] is a large taxonomy derived from categories in Wikipedia by classifying categories as instances or classes and deriving a subsumption hierarchy. The KOG system [43] refines existing Wikipedia infoboxes based on machine learning techniques using both SVMs and a more powerful joint-inference approach expressed in Markov Logic Networks. KYLIN [44] is a system which autonomously extracts structured data from Wikipedia and uses self-

[45] http://www.mediawiki.org/wiki/Alternative_parsers

[46] http://www.mediawiki.org/wiki/Extension_Matrix/data_extraction

[47] http://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/

[48] http://sigwp.org/en/index.php/Wikipedia_Thesaurus

[49] http://sigwp.org/en/

[50] http://baike.baidu.com/

[51] http://www.hudong.com/

[52] http://www.cs.washington.edu/research/knowitall/

supervised linking. [2] was an infobox extraction approach for Wikipedia, which later became the DBpedia project.

9. Conclusions and Future Work

In this system report, we presented an overview on recent advances of the DBpedia community project. The technical innovations described in this article included in particular: (1) the extraction based on the community-curated DBpedia ontology, (2) the live synchronisation of DBpedia with Wikipedia and DBpedia mirrors through update propagation, and (3) the facilitation of the internationalisation of DBpedia. As a result, we demonstrated that DBpedia matured and improved significantly in the last years in particular also in terms of coverage, usability, and data quality.

With DBpedia, we also aim to provide a proof-of-concept and blueprint for the feasibility of large-scale knowledge extraction from crowd-sourced content repositories. There are a large number of further crowd-sourced content repositories and DBpedia already had an impact on their structured data publishing and interlinking. Two examples are Wiktionary with the Wiktionary extraction [18] meanwhile becoming part of DBpedia and *LinkedGeoData* [38], which aims to implement similar data extraction, publishing and linking strategies for *OpenStreetMaps*.

In the future, we see in particular the following directions for advancing the DBpedia project:

Multilingual data integration and fusion. An area, which is still largely unexplored is the integration and fusion between different DBpedia language editions. Non-English DBpedia editions comprise a better and different coverage of local culture. When we are able to precisely identify equivalent, overlapping and complementary parts in different DBpedia language editions, we can reach a significantly increased coverage. On the other hand, comparing the values of a specific property between different language editions will help us to spot extraction errors as well as wrong or outdated information in Wikipedia.

Community-driven data quality improvement. In the future, we also aim to engage a larger community of DBpedia users in feedback loops, which help us to identify data quality problems and corresponding deficiencies of the DBpedia extraction framework. By constantly monitoring the data quality and integrating improvements into the mappings to the DBpedia ontology as well as fixes into the extraction framework, we aim to demonstrate that the Wikipedia community is not only capable to create the largest encyclopedia, but also the most comprehensive and structured knowledge base. With the *DBpedia quality evaluation campaign* [45] we were going a first step in this direction.

Inline extraction. Currently DBpedia extracts information primarily from templates. In the future, we envision to also extract semantic information from typed links. Typed links is a feature of Semantic MediaWiki, which was backported and implemented as a very lightweight extension for MediaWiki[53]. If this extension is deployed at Wikipedia installations, this opens up completely new possibilities for more fine-grained and non-invasive knowledge representations and extraction from Wikipedia.

Collaboration between Wikidata and DBpedia. While DBpedia provides a comprehensive and current view on entity descriptions extracted from Wikipedia, Wikidata offers a variety of factual statements from different sources and dates. One of the richest sources of DBpedia are Wikipedia infoboxes, which are structured but at the same time heterogeneous and non-standardized (thus making the extraction error prone in certain cases). The aim of Wikidata is to populate infoboxes automatically from a centrally managed, high-quality fact database. In this regard, both projects complement each other nicely. In future versions, DBpedia will include more raw data provided by Wikidata and add services such as Linked Data/SPARQL endpoints, RDF dumps, linking and ontology mapping for Wikidata.

Feedback for Wikipedia. A promising prospect is that DBpedia can help to identify misrepresentations, errors and inconsistencies in Wikipedia. In the future, we plan to provide more feedback to the Wikipedia community about the quality of Wikipedia. This can, for instance, be achieved in the form of sanity checks, which are implemented as SPARQL queries on the DBpedia Live endpoint, which identify data quality issues and are executed in certain intervals. For example, a query could check that the birthday of a person must always be before the death day or spot outliers that differ significantly from the range of the majority of the other values. In case a Wikipedia editor makes a mistake or typo when adding such information to a page, this could be automatically identified and provided as feedback to Wikipedians [22].

[53] `http://www.mediawiki.org/wiki/Extension:LightweightRDFa`

Integrate DBpedia and NLP. Despite recent advances (cf. Section 7), there is still a huge potential for employing Linked Data background knowledge in various Natural Language Processing (NLP) tasks. One very promising research avenue in this regard is to employ DBpedia as structured background knowledge for named entity recognition and disambiguation. Currently, most approaches use statistical information such as co-occurrence for named entity disambiguation. However, co-occurrence is not always easy to determine (depends on training data) and update (requires recomputation). With DBpedia and in particular DBpedia Live, we have comprehensive and evolving background knowledge comprising information on the relationship between a large number of real-world entities. Consequently, we can employ this information for deciding to what entity a certain surface form should be mapped.

Acknowledgment

We would like to acknowledge the support of the following people and organisations to DBpedia:

- all Wikipedia contributors
- OpenLink Software for providing and maintaining the server infrastructure for the main DBpedia endpoint.
- Kingsley Idehen for SPARQL and Linked Data hosting and community support
- Paul Kreis for DBpedia development
- Christopher Sahnwaldt for DBpedia development and release management
- Claus Stadler for DBpedia development
- people who helped contributing data for certain parts of the article:
  - ∗ Sindice analysis: Stphane Campinas, Szymon Danielczyk, and Gabriela Vulcu (working at DERI)
  - ∗ Freebase: Shawn Simister, and Tom Morris (working at Freebase)
  - ∗ Instance Data Analysis: Volha Bryl (working at University of Mannheim)

This work was supported by grants from the European Union's 7th Framework Programme provided for the projects LOD2 (GA no. 257943), GeoKnow (GA no. 318159) and Dicode (GA no. 257184).

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2008.

[2] S. Auer and J. Lehmann. What have Innsbruck and Leipzig in common? extracting semantics from wiki content. In *Proceedings of the ESWC (2007)*, volume 4519 of *Lecture Notes in Computer Science*, pages 503–517. Springer, 2007.

[3] C. Becker and C. Bizer. Exploring the geospatial semantic web with DBpedia mobile. *J. Web Sem*, 7(4):278–286, 2009.

[4] D. Bikel, V. Castelli, R. Florian, and D.-J. Han. Entity linking and slot filling through statistical processing and inference rules. In *Proceedings TAC Workshop*, 2009.

[5] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[6] E. Cabrio, J. Cojan, A. P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon. QAKiS: an open domain QA system based on relational patterns. In *ISWC-PD; International Semantic Web Conference (Posters & Demos)*, volume 914 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[7] D. V. Camarda, S. Mazzini, and A. Antonuccio. Lodlive, exploring the web of data. In V. Presutti and H. S. Pinto, editors, *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, pages 197–200. ACM, 2012.

[8] S. Campinas, T. E. Perry, D. Ceccarelli, R. Delbru, and G. Tummarello. Introducing rdf graph summary with application to assisted sparql formulation. In *Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on*, pages 261–266. IEEE, 2012.

[9] D. Damljanovic, M. Agatonovic, and H. Cunningham. Freya: An interactive way of querying linked data using natural language. In *The Semantic Web: ESWC 2011 Workshops*, pages 125–138. Springer, 2012.

[10] O. Erling and I. Mikhailov. RDF support in the virtuoso DBMS. In S. Auer, C. Bizer, C. Müller, and A. V. Zhdanova, editors, *CSSW*, volume 113 of *LNI*, pages 59–68. GI, 2007.

[11] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall. In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM, 2004.

[12] A. García-Silva, M. Jakob, P. N. Mendes, and C. Bizer. Multipedia: enriching DBpedia with multimedia information. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 137–144, New York, NY, USA, 2011. ACM.

[13] A. García-Silva, M. Szomszor, H. Alani, and O. Corcho. Preliminary results in tag disambiguation using DBpedia. In *1st International Workshop in Collective Knowledage Capturing and Representation (CKCaR)*, California, USA, 2009.

[14] R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Bürgle, H. Düwiger, and U. Scheel. Faceted wikipedia search. In W. Abramowicz and R. Tolksdorf, editors, *Business Information Systems, 13th International Conference, BIS 2010, Berlin, Germany, May 3-5, 2010. Proceedings*, volume 47 of

Lecture Notes in Business Information Processing, pages 1–11. Springer, 2010.

[15] P. Heim, T. Ertl, and J. Ziegler. Facet graphs: Complex semantic querying made easy. In *Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010)*, volume 6088 of *LNCS*, pages 288–302, Berlin/Heidelberg, 2010. Springer.

[16] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. Relfinder: Revealing relationships in RDF knowledge bases. In T.-S. Chua, Y. Kompatsiaris, B. Mérialdo, W. Haas, G. Thallinger, and W. Bailer, editors, *Semantic Multimedia, 4th International Conference on Semantic and Digital Media Technologies, SAMT 2009, Graz, Austria, December 2-4, 2009, Proceedings*, volume 5887 of *Lecture Notes in Computer Science*, pages 182–187. Springer, 2009.

[17] P. Heim, S. Lohmann, D. Tsendragchaa, and T. Ertl. Semlens: visual analysis of semantic data with scatter plots and semantic lenses. In C. Ghidini, A.-C. N. Ngomo, S. N. Lindstaedt, and T. Pellegrini, editors, *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 175–178. ACM, 2011.

[18] S. Hellmann, J. Brekle, and S. Auer. Leveraging the crowdsourcing of lexical resources for bootstrapping a linguistic data cloud. In *Proc. of the Joint International Semantic Technology Conference*, 2012.

[19] S. Hellmann, C. Stadler, J. Lehmann, and S. Auer. DBpedia live extraction. In *Proc. of 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, volume 5871 of *Lecture Notes in Computer Science*, pages 1209–1223, 2009.

[20] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell*, 194:28–61, 2013.

[21] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the TAC 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, 2010.

[22] D. Kontokostas, S. Auer, S. Hellmann, J. Lehmann, P. Westphal, R. Cornelissen, and A. Zaveri. Test-driven data quality evaluation for sparql endpoints. In *Submitted to 12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*, 2013.

[23] D. Kontokostas, C. Bratsas, S. Auer, S. Hellmann, I. Antoniou, and G. Metakides. Internationalization of linked data: The case of the greek dbpedia edition. *Web Semantics: Science, Services and Agents on the World Wide Web*, 15(0):51 – 61, 2012.

[24] C. Lagoze, H. V. de Sompel, M. Nelson, and S. Warner. The open archives initiative protocol for metadata harvesting. `http://www.openarchives.org/OAI/openarchivesprotocol.html`, 2008.

[25] J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi. LCC approaches to knowledge base population at TAC 2010. In *Proceedings TAC Workshop*, 2010.

[26] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[27] C.-Y. Lin and E. H. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501, 2000.

[28] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

[29] M. Martin, C. Stadler, P. Frischmuth, and J. Lehmann. Increasing the financial transparency of european commission project funding. *Semantic Web Journal*, Special Call for Linked Dataset descriptions, 2013.

[30] P. N. Mendes, M. Jakob, and C. Bizer. DBpedia for NLP - a multilingual cross-domain knowledge base. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, 2012.

[31] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, Graz, Austria, 2011.

[32] S. M. Meyer, J. Degener, J. Giannandrea, and B. Michener. Optimizing schema-last tuple-store queries in graphd. In A. K. Elmagarmid and D. Agrawal, editors, *SIGMOD Conference*, pages 1047–1056. ACM, 2010.

[33] M. Morsey, J. Lehmann, S. Auer, C. Stadler, and S. Hellmann. DBpedia and the live extraction of structured data from wikipedia. *Program: electronic library and information systems*, 46:27, 2012.

[34] K. Nakayama, M. Pei, M. Erdmann, M. Ito, M. Shirakawa, T. Hara, and S. Nishio. Wikipedia mining: Wikipedia as a corpus for knowledge extraction. In *Annual Wikipedia Conference (Wikimania)*, 2008.

[35] X. Niu, X. Sun, H. Wang, S. Rong, G. Qi, and Y. Yu. Zhishi.me: Weaving chinese linking open data. In *10th International Conference on The semantic web - Volume Part II*, pages 205–220, Berlin, Heidelberg, 2011. Springer-Verlag.

[36] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *Int. J. of Metadata and Semantics and Ontologies*, 3:37–52, Nov. 10 2008.

[37] S. P. Ponzetto and M. Strube. Wikitaxonomy: A large scale knowledge resource. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 751–752. IOS Press, 2008.

[38] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.

[39] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors, *WWW*, pages 697–706. ACM, 2007.

[40] E. Tacchini, A. Schultz, and C. Bizer. Experiments with wikipedia cross-language data fusion. In *Proceedings of the 5th Workshop on Scripting and Development for the Semantic Web, ESWC*. Citeseer, 2009.

[41] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM, 2012.

[42] Z. Wang, J. Li, Z. Wang, and J. Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 21st international conference on World Wide Web*, pages 459–468, New York, NY, USA, 2012. ACM.

[43] F. Wu and D. Weld. Automatically Refining the Wikipedia Infobox Ontology. In *Proceedings of the 17th World Wide Web Conference*, 2008.

[44] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the 16th Conference on Information and Knowledge Management*, pages 41–50. ACM, 2007.

[45] A. Zaveri, D. Kontokostas, M. A. Sherif, L. Bühmann, M. Morsey, S. Auer, and J. Lehmann. User-driven quality evaluation of dbpedia. In *To appear in Proceedings of 9th International Conference on Semantic Systems, I-SEMANTICS '13, Graz, Austria, September 4-6, 2013*. ACM, 2013.

[46] T. Zesch, C. Müller, and I. Gurevych. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May 2008. electronic proceedings.

Appendix

A. Sindice Analysis

The following query was used to analyse the incoming links of all datasets crawled by Sindice including DBpedia. The query runs against the Sindice Summary Graph [8], which contains statistics over all Sindice datasets. Due to the complexity of the query, it was run as a Hadoop job on the Sindice cluster.

```
PREFIX any23:
   <http://vocab.sindice.net/>
PREFIX an:
   <http://vocab.sindice.net/analytics#>
PREFIX dom:
   <http://sindice.com/dataspace/default/
       domain/>
SELECT ?Dataset ?Target_Datatset ?predicate
     SUM(xsd:long(?cardinality)) AS ?total
FROM <http://sindice.com/analytics> {
   ?target any23:domain_uri ?Target_Dataset .

   ?edge an:source ?source ;
         an:target ?target ;
         an:label ?predicate;
         an:cardinality ?cardinality;
         an:publishedIn ?Dataset .

   ?source any23:domain_uri ?Dataset .

FILTER (?Dataset != ?Target_Dataset) }
```

Listing 1: SPARQL query on the Sindice summary graph for obtaining the number of incoming links.