Semantic Web 0 (2017) 1–0
IOS Press

Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO

Editor(s): Amrapali Zaveri, University of Leipzig
Solicited review(s): Zhigang Wang, Beijing Normal University, China; Anonymous; Sebastian Mellor, Newcastle University, U.K.

Michael Färber [*,**], Frederic Bartscherer, Carsten Menne, and Achim Rettinger [***]
Karlsruhe Institute of Technology (KIT), Institute AIFB,
76131 Karlsruhe, Germany

Abstract. In recent years, several noteworthy large, cross-domain, and openly available knowledge graphs (KGs) have been created. These include DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Although extensively in use, these KGs have not been subject to an in-depth comparison so far. In this survey, we provide data quality criteria according to which KGs can be analyzed and analyze and compare the above mentioned KGs. Furthermore, we propose a framework for finding the most suitable KG for a given setting.

Keywords: Knowledge Graph, Linked Data Quality, Data Quality Metrics, Comparison, DBpedia, Freebase, OpenCyc, Wikidata, YAGO

1. Introduction

The vision of the Semantic Web is to publish and query knowledge on the Web in a semantically structured way. According to Guns [23], the term "Semantic Web" had already been used in fields such as Educational Psychology, before it became prominent in Computer Science. Freedman and Reynolds [21], for instance, describe "semantic webbing" as organizing information and relationships in a visual display. Berners-Lee has mentioned his idea of using typed links as vehicle of semantics already since 1989 and proposed it under the term *Semantic Web* for the first time at the INET conference in 1995 [23].

The idea of a Semantic Web was introduced to a wider audience by Berners-Lee in 2001 [10]. According to his vision, the traditional Web as a Web of Documents should be extended to a Web of Data where not only documents and links between documents, but any entity (e.g., a person or organization) and any relation between entities (e.g., *isSpouseOf*) can be represented on the Web.

When it comes to realizing the idea of the Semantic Web, knowledge graphs (KGs) are currently seen as one of the most essential components. The term "knowledge graph" was reintroduced by Google in 2012 [42] and is intended for any graph-based knowledge repository. Since in the Semantic Web RDF graphs are used we use the term **knowledge graph** for any RDF graph. An RDF graph consists of a finite set of RDF triples where each RDF triple (s, p, o) is an ordered set of the following RDF terms: a subject $s \in U \cup B$, a predicate $p \in U$, and an object $o \in U \cup B \cup L$. An RDF term is either a URI $u \in U$, a blank node $b \in B$, or a literal $l \in L$. U, B, and L are infinite sets and pairwise

[*]Corresponding author. E-mail: michael.faerber@kit.edu.
[**]This work was carried out with the support of the German Federal Ministry of Education and Research (BMBF) within the Software Campus project *SUITE* (Grant 01IS12051).
[***] The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 611346.

1570-0844/17/$27.50 © 2017 – IOS Press and the authors. All rights reserved

disjoint. We denote the system that hosts a KG g with h_g.

In this survey, we focus on those KGs having the following aspects:

1. The KGs are freely accessible and freely usable within the Linked Open Data (LOD) cloud.
   Linked Data refers to a set of best practices[1] for publishing and interlinking structured data on the Web, defined by Berners-Lee [8] in 2006. **Linked Open Data** refers to the Linked Data which "can be freely used, modified, and shared by anyone for any purpose."[2] The aim of the Linking Open Data community project[3] is to publish RDF datasets on the Web and to interlink these datasets.
2. The KGs should cover general knowledge (often also called cross-domain or encyclopedic knowledge) instead of knowledge about special domains such as biomedicine.

Thus, out of scope are KGs which are not openly available such as the Google Knowledge Graph[4] and the Google Knowledge Vault [13]. Excluded are also KGs which are only accessible via an API, but which are not provided as dump files (see WolframAlpha[5] and the Facebook Graph[6]) as well as KGs which are not based on Semantic Web standards at all or which are only unstructured or weakly structured knowledge collections (e.g., The World Factbook of the CIA[7]).

For selecting the KGs for analysis, we regarded all datasets which had been registered at the online dataset catalog `http://datahub.io`[8] and which were tagged as "crossdomain". Besides that, we took Wikidata into consideration, since it also fulfilled the above mentioned requirements. Based on that, we selected DBpedia, Freebase, OpenCyc, Wikidata, and YAGO as KGs for our comparison.

In this paper, we give a systematic overview of these KGs in their current versions (as of April 2016) and discuss how the knowledge in these KGs is modeled, stored, and queried. To the best of our knowledge, such a comparison between these widely used KGs has not been presented before. Note that the focus of this survey is not the life cycle of KGs on the Web or in enterprises. We can refer in this respect to [5]. Instead, the focus of our KG comparison is on *data quality*, as this is one of the most crucial aspects when it comes to considering which KG to use in a specific setting.

Furthermore, we provide a KG recommendation framework for users who are interested in using one of the mentioned KGs in a research or industrial setting, but who are inexperienced in which KG to choose for their concrete settings.

The main contributions of this survey are:

1. Based on existing literature on data quality, we provide 34 data quality criteria according to which KGs can be analyzed.
2. We calculate key statistics for the KGs DBpedia, Freebase, OpenCyc, Wikidata, and YAGO.
3. We analyze DBpedia, Freebase, OpenCyc, Wikidata, and YAGO along the mentioned data quality criteria.[9]
4. We propose a framework which enables users to find the most suitable KG for their needs.

The survey is organized as follows:

- In Section 2 we introduce formal definitions used throughout the article.
- In Section 3 we describe the data quality dimensions which we later use for the KG comparison, including their subordinated data quality criteria and corresponding data quality metrics.
- In Section 4 we describe the selected KGs.
- In Section 5 we analyze the KGs using several key statistics and using the data quality metrics introduced in Section 3.
- In Section 6 we present our framework for assessing and rating KGs according to the user's setting.
- In Section 7 we present related work on (linked) data quality criteria and on key statistics for KGs.
- In Section 8 we conclude the survey.

[1] See `http://www.w3.org/TR/ld-bp/`, requested on April 5, 2016.

[2] See `http://opendefinition.org/`, requested on Apr 5, 2016.

[3] See `http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`, requested on Apr 5, 2016.

[4] See `http://www.google.com/insidesearch/features/search/knowledge.html`, requested on Apr 3, 2016.

[5] See `http://products.wolframalpha.com/api/`, requested on Aug 30, 2016.

[6] See `https://developers.facebook.com/docs/graph-api`, requested on Aug 30, 2016.

[7] See `https://www.cia.gov/library/publications/the-world-factbook/`, requested on Aug 30, 2016

[8] This catalog is also used for registering Linked Open Data datasets.

[9] The data and detailed evaluation results for both the key statistics and the metric evaluations are online available at `http://km.aifb.kit.edu/sites/knowledge-graph-comparison/` (requested on Jan 31, 2017).

2. Important Definitions

We define the following sets that are used in formalizations throughout the article. If not otherwise stated, we use the prefixes listed in Listing 1 for indicating namespaces throughout the article.

- C_g denotes the set of **classes** in g: $C_g := \{x \mid (x, \texttt{rdfs:subClassOf}, o) \in g \vee (s, \texttt{rdfs:subClassOf}, x) \in g \vee (x, \texttt{wdt:P279}, o) \in g \vee (s, \texttt{wdt:P279}, x) \in g \vee (x, \texttt{rdf:type}, \texttt{rdfs:Class}) \in g\}$
- An **instance** of a class is a resource which is member of that class. This membership is given by a corresponding instantiation assignment.[10] I_g denotes the set of instances in g: $I_g := \{s \mid (s, \texttt{rdf:type}, o) \in g \vee (s, \texttt{wdt:P31}, o) \in g\}$
- **Entities** are defined as instances which *represent real world objects*. E_g denotes the set of entities in g: $E_g := \{s \mid (s, \texttt{rdf:type}, \texttt{owl:Thing}) \in g \vee (s, \texttt{rdf:type}, \texttt{wdo:Item}) \in g \vee (s, \texttt{rdf:type}, \texttt{freebase:common.topic}) \in g \vee (s, \texttt{rdf:type}, \texttt{cych:Individual}) \in g\}$
- **Relations** (interchangeably used with "properties") are links between RDF terms[11] defined on the schema level (i.e., T-Box). To emphasize this characterization, we also call them **explicitly defined relations**. P_g denotes the set of all those relations in g: $P_g := \{s \mid (s, \texttt{rdf:type}, \texttt{rdf:Property}) \in g \vee (s, \texttt{rdf:type}, \texttt{rdfs:Property}) \in g \vee (s, \texttt{rdf:type}, \texttt{wdo:Property}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:FunctionalProperty}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:InverseFunctionalProperty}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:DatatypeProperty}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:ObjectProperty}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:SymmetricProperty}) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:TransitiveProperty}) \in g\}$
- **Implicitly defined relations** embrace all links used in the KG, i.e., on instance and schema level. We also call them **predicates**. P_g^{imp} denotes the set of all implicitly defined relations in g: $P_g^{imp} := \{p \mid (s, p, o) \in g\}$
- U_g denotes the set of **all URIs** used in g: $U_g := \{x \mid ((x, p, o) \in g \vee (s, x, o) \in g \vee (s, p, x) \in g) \wedge x \in U\}$
- U_g^{local} denotes the set of **all URIs** in g with **local namespace**; i.e., those URIs start with the KG g dedicated prefix (cf. Listing 1).
- Complementary, U_g^{ext} consists of **all URIs** in U_g which are **external** to the KG g which means that h_g is not responsible for resolving those URIs.

Note that knowledge about the KGs which were analyzed for this survey was taken into account when defining these sets. These definitions may not be appropriate for other KGs.

Furthermore, the sets' extensions would be different when assuming a certain semantic (e.g., RDF, RDFS, or OWL-LD). Under the assumption that all entailments under one of these semantics were added to a KG, the definition of each set could be simplified and the extensions would be of larger cardinality. However, for this article we did not derive entailments.

3. Data Quality Assessment w.r.t. KGs

Everybody on the Web can publish information. Therefore, a data consumer does not only face the challenge to find a suitable data source, but is also confronted with the issue that data on the Web can differ very much regarding its quality. Data quality can thereby be viewed not only in terms of accuracy, but in multiple other dimensions. In the following, we introduce concepts regarding the data quality of KGs in the Linked Data context, which are used in the following sections. The data quality dimensions are then exposed in Sections 3.2 – 3.5.

Data quality (DQ) – in the following interchangeably used with *information quality*[12] – is defined by Juran et al. [32] as *fitness for use*. This means that data quality is dependent on the actual use case.

One of the most important and foundational works on data quality is that of Wang et al. [47]. They developed a framework for assessing the data quality of datasets in the database context. In this framework, Wang et al.

[10] See https://www.w3.org/TR/rdf-schema/, requested on Aug 29, 2016.

[11] RDF terms comprise URIs, blank nodes, and literals.

[12] As soon as data is considered w.r.t. usefulness, the data is seen in a specific context. It can, thus, already be regarded as information, leading to the term "information quality" instead of "data quality."

Listing 1: Default prefixes for namespaces used throughout this article.

```
@prefix cc: <http://creativecommons.org/ns#> .
@prefix cyc: <http://sw.opencyc.org/concept/> .
@prefix cych: <http://sw.opencyc.org/2012/05/10/concept/en/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbp: <http://dbpedia.org/property/> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dby: <http://dbpedia.org/class/yago/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix freebase: <http://rdf.freebase.com/ns/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix umbel: <http://umbel.org/umbel/sc/> .
@prefix void: <http://www.w3.org/TR/void#> .
@prefix wdo: <http://www.wikidata.org/ontology#> .
@prefix wdt: <http://www.wikidata.org/entity/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix yago: <http://yago-knowledge.org/resource/> .
```

distinguish between *data quality criteria*, *data quality dimensions*, and *data quality categories*.[13] In the following, we reuse these concepts for our own framework, which has the particular focus on the data quality of KGs in the context of Linked Open Data.

A **data quality criterion** (Wang et al. also call it "data quality attribute") is a particular characteristic of data w.r.t. its quality and can be either subjective or objective. An example of a subjectively measurable data quality criterion is *Trustworthiness on KG level*. An example of an objective data quality criterion is the *Syntactic validity of RDF documents* (see Section 3.2 and [46]).

In order to measure the degree to which a certain data quality criterion is fulfilled for a given KG, each criterion is formalized and expressed in terms of a function with the value range of $[0, 1]$. We call this function the **data quality metric** of the respective data quality criterion.

A **data quality dimension** – in the following just called *dimension* – is a main aspect how data quality can be viewed. A data quality dimension comprises one or several data quality criteria [47]. For instance, the criteria *Syntactic validity of RDF documents*, *Syntactic validity of literals* and *Semantic validity of triples* form the *Accuracy* dimension.

Data quality dimensions and their respective data quality criteria are further grouped into **data quality categories**. Based on empirical studies, Wang et al. specified four categories:

- Criteria of the *category of the intrinsic data quality* focus on the fact that data has quality in its own right.
- Criteria of the *category of the contextual data quality* cannot be considered in general, but must be assessed depending on the application context of the data consumer.
- Criteria of the *category of the representational data quality* reveal in which form the data is available.
- Criteria of the *category of the accessibility data quality* determine how the data can be accessed.

Since its publication, the presented framework of Wang et al. has been extensively used, either in its original version or in an adapted or extended version. Bizer [11] and Zaveri [49] worked on data quality in the Linked Data context. They make the following adaptations on Wang et al.'s framework:

[13] The quality dimensions are defined in [47], the sub-classification into parameters/indicators in [46, p. 354].

- Bizer [11] compared the work of Wang et al. [47] with other works in the area of data quality. He thereby complements the framework with the dimensions *consistency*, *verifiability*, and *offensiveness*.
- Zaveri et al. [49] follow Wang et al. [47], but introduce *licensing* and *interlinking* as new dimensions in the linked data context.

In this article, we use the DQ dimensions as defined by Wang et al. [47] and as extended by Bizer [11] and Zaveri [49]. More precisely, we make the following adaptations on Wang et al.'s framework:

1. *Consistency* is treated by us as separate DQ dimension.
2. *Verifiability* is incorporated within the DQ dimension *Trustworthiness* as criterion *Trustworthiness on statement level*.
3. The *Offensiveness* of KG facts is not considered by us, as it is hard to make an objective evaluation in this regard.
4. We extend the *category of the accessibility data quality* by the dimension *License* and *Interlinking*, as those data quality dimensions get in addition relevant in the Linked Data context.

3.1. Criteria Weighting

When applying our framework to compare KGs, the single DQ metrics can be weighted differently so that the needs and requirements of the users can be taken into account. In the following, we first formalize the idea of weighting the different metrics. We then present the criteria and the corresponding metrics of our framework.

Given are a KG g, a set of criteria $C = \{c_1, ..., c_n\}$, a set of metrics $M = \{m_1, ..., m_n\}$, and a set of weights $W = \{w_1, ..., w_n\}$. Each metric m_i corresponds to the criterion c_i and $m_i(g) \in [0, 1]$ where a value of 0 defines the minimum fulfillment degree of a KG regarding a quality criterion and a value of 1 the maximum fulfillment degree. Furthermore, each criterion c_i is weighted by w_i.

The fulfillment degree $h(g) \in [0, 1]$ of a KG g is then the weighted normalized sum of the fulfillment degrees w.r.t. the criteria $c_1, ..., c_n$:

$$h(g) = \frac{\sum_{i=1}^{n} w_i \, m_i(g)}{\sum_{j=1}^{n} w_j}$$

Based on the quality dimensions introduced by Wang et al. [47], we now present the DQ criteria and metrics as used in our KG comparison. Note that some of the criteria have already been introduced by others as outlined in Section 7.

Note also that our metrics are to be understood as possible ways of how to evaluate the DQ dimensions. Other definitions of the DQ metrics might be possible and reasonable. We defined the metrics along the characteristics of the KGs DBpedia, Freebase, OpenCyc, Wikidata, and YAGO, but kept the definitions as generic as possible. In the evaluations, we then used those metric definitions and applied them, e.g., on the basis of own-created gold standards.

3.2. Intrinsic Category

"Intrinsic data quality denotes that data have quality in their own right." [47] This kind of data quality can therefore be assessed independently from the context. The intrinsic category embraces the three dimensions *Accuracy*, *Trustworthiness*, and *Consistency*, which are defined in the following subsections. The dimensions *Believability*, *Objectivity*, and *Reputation*, which are separate dimensions in Wang et al.'s classification system [47], are subsumed by us under the dimension *Trustworthiness*.

3.2.1. Accuracy

Definition of dimension. Accuracy is "the extent to which data are correct, reliable, and certified free of error" [47].

Discussion. *Accuracy* is intuitively an important dimension of data quality. Previous work on data quality has mainly analyzed only this aspect [47]. Hence, accuracy has often been used as synonym for data quality [39]. Bizer [11] highlights in this context that *Accuracy* is an objective dimension and can only be applied on verifiable statements.

Batini et al. [6] distinguish between syntactic and semantic accuracy: Syntactic accuracy describes the formal compliance to syntactic rules without reviewing whether the value reflects the reality. The semantic accuracy determines whether the value is semantically valid, i.e., whether the value is true. Based on the classification of Batini et al., we can define the metric for *Accuracy* as follows:

Definition of metric. The dimension *Accuracy* is determined by the criteria

- *Syntactic validity of RDF documents*,
- *Syntactic validity of literals*, and

– *Semantic validity of triples*.

The fulfillment degree of a KG g w.r.t. the dimension *Accuracy* is measured by the metrics m_{synRDF}, m_{synLit}, and $m_{semTriple}$, which are defined as follows.

Syntactic validity of RDF documents The syntactic validity of RDF documents is an important requirement for machines to interpret an RDF document completely and correctly. Hogan et al. [29] suggest using standardized tools for creating RDF data. The authors state that in this way normally only little syntax errors occur, despite the complex syntactic representation of RDF/XML.

RDF data can be validated by an RDF validator such as the W3C RDF validator.[14]

$$m_{synRDF}(g) = \begin{cases} 1 & \text{if all RDF documents are valid} \\ 0 & \text{otherwise} \end{cases}$$

Syntactic validity of literals Assessing the syntactic validity of literals means to determine to which degree literal values stored in the KG are syntactically valid. The syntactic validity of literal values depends on the data types of the literals and can be automatically assessed via rules [22,34]. Syntactic rules can be written in the form of regular expressions. For instance, it can be verified whether a literal representing a date follows the ISO 8601 specification. Assuming that L is the infinite set of literals, we can state:

$$m_{synLit}(g) = \frac{|\{(s,p,o) \in g \mid o \in L \wedge synValid(o)\}|}{|\{(s,p,o) \in g \mid o \in L\}|}$$

In case of an empty set in the denominator of the fraction, the metric should evaluate to 1.

Semantic validity of triples The criterion *Semantic validity of triples* is introduced to evaluate whether the statements expressed by the triples (with or without literals) hold *true*. Determining whether a statement is true or false is strictly speaking impossible (see the field of epistemology in philosophy). For evaluating the *Semantic validity of statements*, Bizer et al. [11] note that a triple is semantically correct if it is also available from a trusted source (e.g., Name Authority File), if it is common sense, or if the statement can be measured or perceived by the user directly. Wikidata has similar guidelines implemented to determine whether a fact needs to be sourced.[15]

We measure the *Semantic validity of triples* based on empirical evidence, i.e., based on a reference data set serving as gold standard. We determine the fulfillment degree as the precision that the triples which are in the KG g and in the gold standard GS have the same values. Note that this measurement is heavily depending on the truthfulness of the reference data set.

Formally, let $no_{g,GS} = |\{(s,p,o) \mid (s,p,o) \in g \wedge (x,y,z) \in GS \wedge equi(s,x) \wedge equi(p,y) \wedge equi(o,z))\}|$ be the number of triples in g to which semantically corresponding triples in the gold standard GS exist. Let $no_g = |\{(s,p,o) \mid (s,p,o) \in g \wedge (x,y,z) \in GS \wedge equi(s,x) \wedge equi(p,y)\}|$ be the number of triples in g where the subject-relation-pairs (s,p) are semantically equivalent to subject-relation-pairs (x,y) in the gold standard. Then we can state:

$$m_{semTriple}(g) = \frac{no_{g,GS}}{no_g}$$

In case of an empty set in the denominator of the fraction, the metric should evaluate to 1.

3.2.2. Trustworthiness

Definition of dimension. Trustworthiness is defined as "the degree to which the information is accepted to be correct, true, real, and credible" [49]. We define it as a collective term for *believability*, *reputation*, *objectivity*, and *verifiability*. These aspects were defined by Wang et al. [47] and Naumann [39] as follows:

– **Believability**: Believability is "the extent to which data are accepted or regarded as true, real, and credible" [47].
– **Reputation**: Reputation is "the extent to which data are trusted or highly regarded in terms of their source or content" [47].
– **Objectivity**: Objectivity is "the extent to which data are unbiased (unprejudiced) and impartial" [47].
– **Verifiability**: Verifiability is "the degree and ease with which the data can be checked for correctness" [39].

[14] See `http://www.w3.org/RDF/Validator`, requested on Feb 29, 2016.

[15] See `https://www.wikidata.org/wiki/Help:Sources`, requested on Sep 8, 2016.

Discussion. In summary, *believability* considers the subject (data consumer) side; *reputation* takes the general, social view on trustworthiness; *objectivity* considers the object (data provider) side, while *verifiability* focuses on the possibility of verification.

Trustworthiness has been discussed as follows:

- **Believability**: According to Naumann [39], *believability* is the "expected accuracy" of a data source.
- **Reputation**: The essential difference of *believability* to *accuracy* is that for *believability*, data is trusted without verification [11]. Thus, *believability* is closely related to the *reputation* of a dataset.
- **Objectivity**: According to Naumann [39], the *objectivity* of a data source is strongly related to the *verifiability*: The more verifiable a data source or statement is, the more objective it is. The authors of this article would not go so far, since also biased statements could be verifiable.
- **Verifiability**: Heath et al. [26] emphasize that it is essential for trustworthy applications to be able to verify the origin of data.

Definition of metric. We define the metric for the data quality dimension *Trustworthiness* as a combination of trustworthiness metrics on both KG and statement level. *Believability* and *reputation* are thereby covered by the DQ criterion *Trustworthiness on KG level* (metric $m_{graph}(h_g)$), while *objectivity* and *verifiability* are covered by the DQ criteria *Trustworthiness on statement level* (metric $m_{fact}(g)$) and *Indicating unknown and empty values* (metric $m_{NoVal}(g)$). Hence, the fulfillment degree of a KG g w.r.t. the dimension *Trustworthiness* is measured by the metrics m_{graph}, m_{fact}, and m_{NoVal}, which are defined as follows.

Trustworthiness on KG level The measure of *Trustworthiness on KG level* exposes a basic indication about the trustworthiness of the KG. In this assessment, the method of data curation as well as the method of data insertion is taken into account. Regarding the method of data curation, we distinguish between manual and automated methods. Regarding the data insertion, we can differentiate between: 1. whether the data is entered by experts (of a specific domain), 2. whether the knowledge comes from volunteers contributing in a community, and 3. whether the knowledge is extracted automatically from a data source. This data source can itself be either structured, semi-structured, or un-structured. We assume that a closed system, where experts or other registered users feed knowledge into a system, is less vulnerable to harmful behavior of users than an open system, where data is curated by a community. Therefore, we assign the values of the metric for *Trustworthiness on KG level* as follows:

$$m_{graph}(h_g) = \begin{cases} 1 & \text{manual data curation, manual data insertion in a closed system} \\ 0.75 & \text{manual data curation and insertion, both by a community} \\ 0.5 & \text{manual data curation, data insertion by community or data insertion by automated knowledge extraction} \\ 0.25 & \text{automated data curation, data insertion by automated knowledge extraction from structured data sources} \\ 0 & \text{automated data curation, data insertion by automated knowledge extraction from unstructured data sources} \end{cases}$$

Note that all proposed DQ metrics should be seen as suggestions of how to formulate DQ metrics. Hence, other numerical values and other classification schemes (e.g., for $m_{graph}(h_g)$) might be taken for defining the DQ metrics.

Trustworthiness on statement level The fulfillment of *Trustworthiness on statement level* is determined by an assessment whether a provenance vocabulary is used. By means of a provenance vocabulary, the source of statements can be stored. Storing source information is an important precondition to assess statements easily w.r.t. semantic validity. We distinguish between provenance information provided for triples and provenance information provided for resources.

The most widely used ontologies for storing provenance information are the Dublin Core Metadata terms[16] with properties such as `dcterms:provenance` and `dcterms:source` and the W3C PROV ontology[17] with properties such as `prov:wasDerivedFrom`.

[16] See `http://purl.org/dc/terms/`, requested on Feb 4, 2017.

[17] See `https://www.w3.org/TR/prov-o/`, requested on Dec 27, 2016.

$$m_{fact}(g) = \begin{cases} 1 & \text{provenance on statement level is used} \\ 0.5 & \text{provenance on resource level is used} \\ 0 & \text{otherwise} \end{cases}$$

Indicating unknown and empty values If the data model of the considered KG supports the representation of unknown and empty values, more complex statements can be represented. For instance, empty values allow to represent that a person has no children and unknown values allow to represent that the birth date of a person in not known. This kind of higher explanatory power of a KG increases the trustworthiness of the KG.

$$m_{NoVal}(g) = \begin{cases} 1 & \text{unknown and empty values are used} \\ 0.5 & \text{either unknown or empty values are used} \\ 0 & \text{otherwise} \end{cases}$$

3.2.3. Consistency

Definition of dimension. Consistency implies that "two or more values [in a dataset] do not conflict each other" [37].

Discussion. Due to the high variety of data providers in the Web of Data, a user must expect data inconsistencies. Data inconsistencies may be caused by (i) different information providers, (ii) different levels of knowledge, and (iii) different views of the world [11].

In OWL, restrictions can be introduced to ensure consistent modeling of knowledge to some degree. The OWL schema restrictions can be divided into class restrictions and relation restrictions [7].

Class restrictions refer to classes. For instance, one can specify via `owl:disjointWith` that two classes have no common instance.

Relation restrictions refer to the usage of relations. They can be classified into value constraints and cardinality constraints.

Value constraints determine the range of relations. `owl:someValuesFrom`, for instance, specifies that at least one value of a relation belongs to a certain class. If the expected data type of a relation is specified via `rdfs:range`, we also consider this as relation restriction.

Cardinality constraints limit the number of times a relation may exist per resource. Via `owl:Functional property` and `owl:InverseFunctionalProperty`, global cardinality constraints can be specified. Functional relations permit at most one value per resource (e.g., the birth date of a person). Inverse functional relations specify that a value should only occur once per resource. This means that the subject is the only resource linked to the given object via the given relation.

Definition of metric. We can measure the data quality dimension *Consistency* by means of (i) whether schema constraints are checked during the insertion of new statements into the KG and (ii) whether already existing statements in the KG are consistent to specified class and relation constraints. The fulfillment degree of a KG g w.r.t. the dimension *consistency* is measured by the metrics $m_{checkRestr}$, $m_{conClass}$, and $m_{conRelat}$, which are defined as follows.

Check of schema restrictions during insertion of new statements Checking the schema restrictions during the insertion of new statements can help to reject facts that would render the KG inconsistent. Such simple checks are often done on the client side in the user interface. For instance, the application checks whether data with the right data type is inserted. Due to the dependency to the actual inserted data, the check needs to be custom-designed. Simple rules are applicable, however, inconsistencies can still appear if no suitable rules are available. Examples of consistency checks are: checking the expected data types of literals; checking whether the entity to be inserted has a valid entity type (i.e., checking the `rdf:type` relation); checking whether the assigned classes of the entity are disjoint, i.e., contradicting each other (utilizing `owl:disjointWith` relations).

$$m_{checkRestr}(h_g) = \begin{cases} 1 & \text{schema restrictions are checked} \\ 0 & \text{otherwise} \end{cases}$$

Consistency of statements w.r.t. class constraints This metric is intended to measure the degree to which the instance data is consistent with the class restrictions (e.g., `owl:disjointWith`) specified on the schema level.

In the following, we limit ourselves to the class constraints given by all `owl:disjointWith` statements defined on the schema level of the considered KG. I.e., let CC be the set of all class constraints, defined as $CC := \{(c_1, c_2) \mid (c_1, \texttt{owl:dis-}$

jointWith, c_2) $\in g$}[18]. Furthermore, let $c_g(e)$ be the set of all classes of instance e in g, defined as $c_g(e) = \{c \mid (e, \texttt{rdf:type}, c) \in g\}$. Then we define $m_{conClass}(g)$ as follows:

$$m_{conClass}(g) = \frac{|\{(c_1, c_2) \in CC \mid \neg\exists e : (c_1 \in c_g(e) \wedge c_2 \in c_g(e))\}|}{|\{(c_1, c_2) \in CC\}|}$$

In case of an empty set of class constraints CC, the metric should evaluate to 1.

Consistency of statements w.r.t. relation constraints The metric for this criterion is intended for measuring the degree to which the instance data is consistent with the relation restrictions (e.g., indicated via `rdfs:range` and `owl:FunctionalProperty`) specified on the schema level. We evaluate this criterion by averaging over the scores obtained from single metrics $m_{conRelat,i}$ indicating the consistency of statements w.r.t. different relation constraints:

$$m_{conRelat}(g) = \frac{1}{n}\sum_{i=1}^{n} m_{conRelat,i}(g)$$

In case of evaluating the consistency of instance data concretely w.r.t. given `rdfs:range` and `owl:FunctionalProperty` statements,[19] we can state

$$m_{conRelat}(g) = \frac{m_{conRelatRg}(g) + m_{conRelatFct}(g)}{2}$$

Let R_r be the set of all `rdfs:range` constraints,

$$R_r := \{(p, d) \mid (p, \texttt{rdfs:range}, d) \in g \wedge isDatatype(d)\}$$

and R_f be the set of all `owl:FunctionalProperty` constraints,

$$R_f := \{(p, d) \mid (p, \texttt{rdf:type}, \texttt{owl:FunctionalProperty}) \in g \wedge (p, \texttt{rdfs:range}, d) \in g \wedge isDatatype(d)\}$$

Then we can define the metrics $m_{conRelatRg}(g)$ and $m_{conRelatFct}(g)$ as follows:

$$m_{conRelatRg}(g) = \frac{|\{(s, p, o) \in g \mid \exists (p, d) \in R_r : datatype(o) \neq d\}|}{|\{(s, p, o) \in g \mid \exists (p, d) \in R_r\}|}$$

$$m_{conRelatFct}(g) = \frac{|\{(s, p, o) \in g | \exists (p, d) \in R_f : \neg\exists (s, p, o_2) \in g : o \neq o_2\}|}{|\{(s, p, o) \in g \mid \exists (p, d) \in R_f\}|}$$

In case of an empty set of relation constraints (R_r or R_f), the respective metric should evaluate to 1.

3.3. Contextual Category

Contextual data quality "highlights the requirement that data quality must be considered within the context of the task at hand" [47]. This category contains the three dimensions (i) *Relevancy*, (ii) *Completeness*, and (iii) *Timeliness*. Wang et al.'s further dimensions in this category, *appropriate amount of data* and *value-added*, are considered by us as being part of the dimension *Completeness*.

3.3.1. Relevancy

Definition of dimension. Relevancy is "the extent to which data are applicable and helpful for the task at hand" [47].

Discussion. According to Bizer [11], *Relevancy* is an important quality dimension, since the user is confronted with a variety of potentially relevant information on the Web.

Definition of metric. The dimension *Relevancy* is determined by the criterion *Creating a ranking of statements*.[20] The fulfillment degree of a KG g w.r.t. the dimension *Relevancy* is measured by the metric $m_{Ranking}$, which is defined as follows.

[18] Implicit restrictions which can be deducted from the class hierarchy, e.g., that a restriction for `dbo:Animal` counts also for `dbo:Mammal`, a subclass of `dbo:Animal`, are not considered by us here.

[19] We chose those relations (and, for instance, not `owl:InverseFunctionalProperty`), as only those relations are used by more than half of the considered KGs.

[20] We do not consider the relevancy of literals, as there is no ranking of literals provided for the considered KGs.

Creating a ranking of statements By means of this criterion one can determine whether the KG supports a ranking of statements by which the relative relevance of statements among other statements can be expressed. For instance, given the Wikidata entity "Barack Obama" (`wdt:Q76`) and the relation "position held" (`wdt:P39`), "President of the United States of America" (`wdt:Q11696`) has a "preferred rank" (`wdo:PreferredRank`) (until 2017), while older positions which he holds no more are ranked as "normal rank" (`wdo:NormalRank`).

$$m_{Ranking}(g) = \begin{cases} 1 & \text{ranking of statements supported} \\ 0 & \text{otherwise} \end{cases}$$

Note that this criterion refers to a characteristic of the KG and not to a characteristic of the system that hosts the KG.

3.3.2. Completeness

Definition of dimension. Completeness is "the extent to which data are of sufficient breadth, depth, and scope for the task at hand" [47].

We include the following two aspects in this dimension, which are separate dimensions in Wang et al.'s framework:

- *Appropriate amount of data:* Appropriate amount of data is "the extent to which the quantity or volume of available data is appropriate" [47].
- *Value-added:* Value-added is "the extent to which data are beneficial and provide advantages from their use" [47].

Discussion. Pipino et al. [40] divide *Completeness* into

1. *Schema completeness*, i.e., the extent to which classes and relations are not missing,
2. *Column completeness*, i.e., the extent to which values of relations on instance level – i.e., facts – are not missing, and
3. *Population completeness*, i.e., the extent to which entities are not missing.

The *Completeness* dimension is context-dependent and therefore belongs to the contextual category, because the fact that a KG is seen as complete depends on the use case scenario, i.e., on the given KG and on the information need of the user. As exemplified by Bizer [11], a list of German stocks is complete for an investor who is interested in German stocks, but it is not complete for an investor who is looking for an overview of European stocks. The completeness is, hence, only assessable by means of a concrete use case at hand or with the help of a defined gold standard.

Definition of metric. We follow the above-mentioned distinction of Pipino et al. [40] and determine *Completeness* by means of the criteria *Schema completeness*, *Column completeness*, and *Population completeness*.

The fulfillment degree of a KG g w.r.t. the dimension *Completeness* is measured by the metrics $m_{cSchema}$, m_{cCol}, and m_{cPop}, which are defined as follows.

Schema completeness By means of the criterion *Schema completeness*, one can determine the completeness of the schema w.r.t. classes and relations [40]. The schema is assessed by means of a gold standard. This gold standard consists of classes and relations which are relevant for the use case. For evaluating cross-domain KGs, we use as gold standard a typical set of cross-domain classes and relations. It comprises (i) basic classes such as people and locations in different granularities and (ii) basic relations such as birth date and number of inhabitants. We define the schema completeness $m_{cSchema}$ as the ratio of the number of classes and relations of the gold standard existing in g, no_{clatg}, and the number of classes and relations in the gold standard, no_{clat}.

$$m_{cSchema}(g) = \frac{no_{clatg}}{no_{clat}}$$

Column completeness In the traditional database area (with fixed schema), by means of the *Column completeness* criterion one can determine the degree by which the relations of a class, which are defined on the schema level (each relation has one column), exist on the instance level [40]. In the Semantic Web and Linked Data context, however, we cannot presume any fixed relational schema on the schema level. The set of possible relations for the instances of a class is given "at runtime" by the set of used relations for the instances of this class. Therefore, we need to modify this criterion as already proposed by Pipino et al. [40]. In the updated version, by means of the criterion *Column completeness* one can determine the degree by which the instances of a class use the same relations, averaged over all classes.

Formally, we define the *Column completeness* metric $m_{cCol}(g)$ as the ratio of the number of instances having class k and a value for the relation r, no_{kp}, to the number of all instances having class k, no_k. By averaging over all class-relation-pairs which occur on

instance level, we obtain a fulfillment degree regarding the whole KG:

$$m_{cCol}(g) = \frac{1}{|H|} \sum_{(k,p)\in H} \frac{no_{kp}}{no_k}$$

We thereby let $H = \{(k,p) \in (K \times P) \mid \exists k \in C_g \wedge \exists (x,p,o) \mid p \in P_g^{imp} \wedge (x, \texttt{rdf:type}, k)\}$ be the set of all combinations of the considered classes, $K = \{k_1, ..., k_n\}$, and considered relations, $P = \{p_1, ..., p_m\}$.

Note that there are also relations which are dedicated to the instances of a specific class, but which do not need to exist for all instances of that class. For instance, not all people need to have a relation `:hasChild` or `:deathDate`.[21] For measuring the *Column completeness*, we selected only those relations for an assessment where a value of the relation typically exists for all given instances.

Population completeness The *Population completeness* metric determines the extent to which the considered KG covers a basic population [40]. The assessment of the KG completeness w.r.t. a basic population is performed by means of a gold standard, which covers both well-known entities (called "short head", e.g., the n largest cities in the world according to the number of inhabitants) and little-known entities (called "long tail"; e.g., municipalities in Germany). We take all entities contained in our gold standard equally into account.

Let GS be the set of entities in the gold standard. Then we can define:

$$m_{cPop}(g) = \frac{|\{e|e \in GS \wedge e \in E_g\}|}{|\{e|e \in GS\}|}$$

3.3.3. Timeliness

Definition of dimension. Timeliness is "the extent to which the age of the data is appropriate for the task at hand" [47].

Discussion. *Timeliness* does not describe the creation date of a statement, but instead the time range since the last update or the last verification of the statement [39]. Due to the easy way of publishing data on the Web, data sources can be kept easier up-to-date than traditional isolated data sources. This results in advantages to the consumer of Web data [39]. How *Timeliness* is measured depends on the application context: For some situations years are sufficient, while in other situations one may need days [39].

Definition of metric. The dimension *timeliness* is determined by the criteria *Timeliness frequency of the KG*, *Specification of the validity period*, and *Specification of the modification date of statements*.

The fulfillment degree of a KG g w.r.t. the dimension *Timeliness* is measured by the metrics m_{Freq}, $m_{Validity}$, and m_{Change}, which are defined as follows.

Timeliness frequency of the KG The criterion *Timeliness frequency of the KG* indicates how fast the KG is updated. We consider the KG RDF export here and differentiate between continuous updates, where the updates are always performed immediately, and discrete KG updates, where the updates take place in discrete time intervals. In case the KG edits are available online immediately but the RDF export files are available in discrete, varying updating intervals, we consider the online version of the KG, since in the context of Linked Data it is sufficient that URIs are dereferenceable.

$$m_{Freq}(g) = \begin{cases} 1 & \text{continuous updates} \\ 0.5 & \text{discrete periodic updates} \\ 0.25 & \text{discrete non-periodic updates} \\ 0 & \text{otherwise} \end{cases}$$

Specification of the validity period of statements Specifying the validity period of statements enables to temporally limit the validity of statements. By using this criterion, we measure whether the KG supports the specification of starting and maybe end dates of statements by means of providing suitable forms of representation.

$$m_{Validity}(g) = \begin{cases} 1 & \text{specification of validity period supported} \\ 0 & \text{otherwise} \end{cases}$$

Specification of the modification date of statements The modification date discloses the point in time of the last verification of a statement. The modification date is typically represented via the relations `schema:dateModified` and `dcterms:modified`.

$$m_{Change}(g) = \begin{cases} 1 & \text{specification of modification dates for statements supported} \\ 0 & \text{otherwise} \end{cases}$$

[21] For an evaluation about the prediction which relations are of this nature, see [1].

3.4. Representational Data Quality

Representational data quality "contains aspects related to the format of the data [...] and meaning of data" [47]. This category contains the two dimensions (i) *Ease of understanding* (i.e., regarding the human-readability) and (ii) *Interoperability* (i.e., regarding the machine-readability). The dimensions *Interpretability*, *Representational consistency* and *Concise representation* as in addition proposed by Wang et al. [47] are considered by us as being a part of the dimension *Interoperability*.

3.4.1. Ease of Understanding

Definition of dimension. The ease of understanding is "the extent to which data are clear without ambiguity and easily comprehended" [47].

Discussion. This dimension focuses on the understandability of a data source by a human data consumer. In contrast, the dimension *Interoperability* focuses on technical aspects. The understandability of a data source (here: KG) can be improved by things such as descriptive labels and literals in multiple languages.

Definition of metric. The dimension *understandability* is determined by the criteria *Description of resources*, *Labels in multiple languages*, *Understandable RDF serialization*, and *Self-describing URIs*. The fulfillment degree of a KG g w.r.t. the dimension *Consistency* is measured by the metrics m_{Descr}, m_{Lang}, m_{uSer}, and m_{uURI}, which are defined as follows.

Description of resources Heath et al. [26,30] suggest to describe resources in a human-understandable way, e.g., via `rdfs:label` or `rdfs:comment`. Within our framework, the criterion is measured as follows: Given a sample of resources, we divide the number of resources in the KG for which at least one label or one description is provided, (e.g., via `rdfs:label`, `rdfs:comment`, or `schema:description`) by the number of all considered resources in the local namespace:

$$m_{Descr}(g) = |\{u | u \in U_g^{local} \wedge \exists (u,p,o) \in g : p \in P_{lDesc}\}| / |\{u | u \in U_g^{local}\}|$$

P_{lDesc} is the set of implicitly used relations in g indicating that the value is a label or description (e.g., $P_{lDesc} = \{$`rdfs:label`, `rdfs:comment`$\}$).

Beschreibung). Darüber hinaus ist das Ergebnis der Evaluation auf Basis der Entitäten interessant -> DBpedia weicht deutlich ab, da manche Entitäten (Intermediate-Node-Mapping) keine rdfs:label haben. Folglich würde ich die Definition der Metrik allgemein halten (beschränkt auf proprietäre Ressourcen, d.h. im selben Namespace), die Evaluation jedoch nur anhand der Entitäten machen.

Labels in multiple languages Resources in the KG are described in a human-readable way via labels, e.g., via `rdfs:label` or `skos:prefLabel`.[22] The characteristic feature of `skos:prefLabel` is that this kind of label should be used per resource at most once; in contrast, `rdfs:label` has no cardinality restrictions, i.e., it can be used several times for a given resource. Labels are usually provided in English as the "basic language." The now introduced metric for the criterion *Labels in multiple languages* determines whether labels in other languages than English are provided in the KG.

$$m_{Lang}(g) = \begin{cases} 1 & \text{Labels provided in English and at least one other language} \\ 0 & \text{otherwise} \end{cases}$$

Understandable RDF serialization RDF/XML is the recommended RDF serialization format of the W3C. However, due to its syntax RDF/XML documents are hard to read for humans. The understandability of RDF data by humans can be increased by providing RDF in other, more human-understandable serialization formats such as N3, N-Triple, and Turtle. We measure this criterion by measuring the supported serialization formats during the dereferencing of resources.

$$m_{uSer}(h_g) = \begin{cases} 1 & \text{Other RDF serializations than RDF/XML available} \\ 0 & \text{otherwise} \end{cases}$$

Note that conversions from one RDF serialization format into another are easy to perform.

Self-describing URIs Descriptive URIs contribute to a better human-readability of KG data. Sauermann et al.[23] recommend to use short, memorable URIs in the Semantic Web context, which are easier understandable and memorable by humans compared to *opaque URIs*[24]

[22] Using the namespace `http://www.w3.org/2004/02/skos/core#`.

[23] See `https://www.w3.org/TR/cooluris`, requested on Mar 1, 2016.

[24] For an overview of URI patterns see `https://www.w3.org/community/bpmlod/wiki/Best_practises_-_previous_notes`, requested on Dec 27, 2016.

such as `wdt:Q1040`. The criterion *Self-describing URIs* is dedicated to evaluate whether self-describing URIs or generic IDs are used for the identification of resources.

$$m_{uURI}(g) = \begin{cases} 1 & \text{self-describing URIs always used} \\ 0.5 & \text{self-describing URIs partly used} \\ 0 & \text{otherwise} \end{cases}$$

3.4.2. Interoperability

Interoperability is another dimension of the representational data quality category and subsumes Wang et al.'s aspects *interpretability*, *representational consistency*, and *concise representation*.

Definition of dimension. We define *Interoperability* along the subsumed dimensions of Wang et al.:

- **Interpretability:** Interpretability is "the extent to which data are in appropriate language and units and the data definitions are clear" [47].
- **Representational consistency:** Representational consistency is "the extent to which data are always presented in the same format and are compatible with previous data" [47].
- **Concise representation:** Concise representation is "the extent to which data are compactly represented without being overwhelming" [47].

Discussion regarding interpretability. In contrast to the dimension understandability, which focuses on the understandability of RDF KG data towards the user as data consumer, interpretability focuses on the representation forms of information in the KG from a technical perspective. An example is the consideration whether blank nodes are used. According to Heath et al. [26], blank nodes should be avoided in the Linked Data context, since they complicate the integration of multiple data sources and since they cannot be linked by resources of other data sources.

Discussion regarding representational consistency. In the context of Linked Data, it is best practice to reuse existing vocabulary for the creation of own RDF data. In this way, less data needs to be prepared for being published as Linked Data [26].

Discussion regarding concise representation. Heath et al. [26] made the observation that the RDF features (i) RDF reification,[25] (ii) RDF collections and RDF container, and (iii) blank nodes are not very widely used in the Linked Open Data context. Those features should be avoided according to Heath et al. in order to simplify the processing of data on the client side. Even the querying of the data via SPARQL may get complicated if RDF reification, RDF collections, and RDF container are used. We agree on that, but also point out that reification (implemented via RDF standard reification, n-ary relations, singleton properties, or named graphs) is inevitably necessary for making statements about statements.

Definition of metric. The dimension *Interoperability* is determined via the following criteria:

- *Avoiding blank nodes and RDF reification*
- *Provisioning of several serialization formats*
- *Using external vocabulary*
- *Interoperability of proprietary vocabulary*

The fulfillment degree of a KG g w.r.t. the dimension *Interoperability* is measured by the metrics m_{Reif}, $m_{iSerial}$, m_{exVoc}, and $m_{propVoc}$, which are defined as follows.

Avoiding blank nodes and RDF reification Using RDF blank nodes, RDF reification, RDF container, and RDF lists is often considered as ambivalent: On the one hand, these RDF features are not very common and they complicate the processing and querying of RDF data [30,26]. On the other hand, they are necessary in certain situations, e.g., when statements about statements should be made. We measure the criterion by evaluating whether blank nodes and RDF reification are used.

$$m_{Reif}(g) = \begin{cases} 1 & \text{no blank nodes and no RDF reification} \\ 0.5 & \text{either blank nodes or RDF reification} \\ 0 & \text{otherwise} \end{cases}$$

Provisioning of several serialization formats The interpretability of RDF data of a KG is increased if be-

[25] In the literature, it is often not differentiated between "reification" in the general sense and "reification" in the sense of the specific proposal described in the RDF standard (Brickley, D., Guha, R. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, online available at `http://www.w3.org/TR/rdf-schema/`, requested on Sep 2, 2016.). For more information about reification and its implementation possibilities, we can refer the reader to [27]. In this article, we use the term "reification" by default for the general sense and "standard reification" or "RDF reification" for referring to the modeling of reification according to the RDF standard.

sides the serialization standard RDF/XML further serialization formats are supported for URI dereferencing.

$$m_{iSerial}(h_g) = \begin{cases} 1 & \text{RDF/XML and further formats are supported} \\ 0.5 & \text{only RDF/XML is supported} \\ 0 & \text{otherwise} \end{cases}$$

Using external vocabulary Using a common vocabulary for representing and describing the KG data allows to represent resources and relations between resources in the Web of Data in a unified way. This increases the interoperability of data [30,26] and allows a comfortable data integration. We measure the criterion of using an external vocabulary by setting the number of triples with external vocabulary in predicate position to the number of all triples in the KG:

$$m_{extVoc}(g) = \frac{|\{(s,p,o)|(s,p,o) \in g \wedge p \in P_g^{external}\}|}{|\{(s,p,o) \in g\}|}$$

Interoperability of proprietary vocabulary Linking on schema level means to link the proprietary vocabulary to external vocabulary. Proprietary vocabulary are classes and relations which were defined in the KG itself. The interlinking to external vocabulary guarantees a high degree of interoperability [26]. We measure the interlinking on schema level by calculating the ratio to which classes and relations have at least one equivalency link (e.g., `owl:sameAs`, `owl:equivalentProperty`, or `owl:equivalentClass`) to classes and relations, respectively, of other data sources.

$$m_{propVoc}(g) = |\{x \in P_g \cup C_g | \exists (x,p,o) \in g : (p \in P_{eq} \wedge (o \in U \wedge o \in U_g^{ext}))\}| / |P_g \cup C_g|$$

where $P_{eq} = \{$`owl:sameAs`, `owl:equivalentProperty`, `owl:equivalenClass`$\}$ and U_g^{ext} consists of all URIs in U_g which are external to the KG g which means that h_g is not responsible for resolving these URIs.

3.5. Accessibility Category

Accessibility data quality refers to aspects on how data can be accessed. This category contains the three dimensions

- *Accessibility*,
- *Licensing*, and
- *Interlinking*.

Wang's dimension *access security* is considered by us as being not relevant in the Linked Open Data context, as we only take open data sources into account.

In the following, we go into details of the mentioned data quality dimensions:

3.5.1. Accessibility

Definition of dimension. Accessibility is "the extent to which data are *available* or *easily* and *quickly* retrievable" [47].

Discussion. Wang et al.'s definition of *Accessibility* contains the aspects *availability*, *response time*, and *data request*. They are defined as follows:

1. *Availability* "of a data source is the probability that a feasible query is correctly answered in a given time range" [39].
   According to Naumann [39], the availability is an important quality aspect for data sources on the Web, since in case of integrated systems (with federated queries) usually all data sources need to be available in order to execute the query. There can be different influencing factors regarding the availability of data sources, such as the day time, the worldwide distribution of servers, the planed maintenance work, and the caching of data. Linked Data sources can be available as SPARQL endpoints (for performing complex queries on the data) and via HTTP URI dereferencing. We need to consider both possibilities for this DQ dimension.
2. *Response time* characterizes the delay between the point in time when the query was submitted and the point in time when the query response is received [11].
   Note that the response time is dependent on empirical factors such as the query, the size of the indexed data, the data structure, the used triple store, the hardware, and so on. We do not consider the response time in our evaluations, since obtaining a comprehensive result here is hard.
3. In the context of Linked Data, *data requests* can be made (i) on SPARQL endpoints, (ii) on RDF dumps (export files), and (iii) on Linked Data APIs.

Definition of metric. We define the metric for the dimension *Accessibility* by means of metrics for the following criteria:

– *Dereferencing possibility of resources*
– *Availability of the KG*
– *Provisioning of public SPARQL endpoint*
– *Provisioning of an RDF export*
– *Support of content negotiation*
– *Linking HTML sites to RDF serializations*
– *Provisioning of KG metadata*

The fulfillment degree of a KG g w.r.t. the dimension *Accessibility* is measured by the metrics m_{Deref}, m_{Avai}, m_{SPARQL}, m_{Export}, m_{Negot}, $m_{HTMLRDF}$, and m_{Meta} , which are defined as follows.

Dereferencing possibility of resources One of the Linked Data principles [9] is the dereferencing possibility of resources: URIs must be resolvable via HTTP requests and useful information should be returned thereby. We assess the dereferencing possibility of resources in the KG by analyzing for each URI in the sample set (here: all URIs U_g) the HTTP response status code and by evaluating whether RDF data is returned. A successful dereferencing of resources is given if HTTP status code 200 and an RDF document is returned.

$$m_{Deref}(h_g) = \frac{|dereferencable(U_g)|}{|U_g|}$$

Availability of the KG The *Availability of the KG* criterion indicates the uptime of the KG. It is an essential criterion in the context of Linked Data, since in case of an integrated or federated query mostly all data sources need to be available [39]. We measure the availability of a KG by monitoring the ability of dereferencing URIs over a period of time. This monitoring process can be done with the help of a monitoring tool such as Pingdom.[26]

$$m_{Avai}(h_g) = \frac{\text{Number of successful requests}}{\text{Number of all requests}}$$

Provisioning of public SPARQL endpoint SPARQL endpoints allow the user to perform complex queries (including potentially many instances, classes, and relations) on the KG. This criterion here indicates whether an official SPARQL endpoint is publicly available. There might be additional restrictions of this SPARQL endpoint such as a maximum number of requests per time slice or a maximum runtime of a query. However, we do not measure these restrictions here.

$$m_{SPARQL}(h_g) = \begin{cases} 1 & \text{SPARQL endpoint publicly available} \\ 0 & \text{otherwise} \end{cases}$$

Provisioning of an RDF export If there is no public SPARQL endpoint available or the restrictions of this endpoint are so strict that the user does not use it, an RDF export dataset (RDF dump) can often be used. This dataset can be used to set up a local, private SPARQL endpoint. The criterion here indicates whether an RDF export dataset is officially available:

$$m_{Export}(h_g) = \begin{cases} 1 & \text{RDF export available} \\ 0 & \text{otherwise} \end{cases}$$

Support of content negotiation Content negotiation (CN) allows that the server returns RDF documents during the dereferencing of resources in the desired RDF serialization format. The HTTP protocol allows the client to specify the desired content type (e.g., RDF/XML) in the HTTP request and the server to specify the returned content type in the HTTP response header (e.g., `application/rdf+xml`). In this way, the desired and the provided content type are matched as far as possible. It can happen that the server does not provide the desired content type. Moreover, it may happen that the server returns an incorrect content type. This may lead to the fact that serialized RDF data is not processed further. An example is RDF data which is declared as `text/plain` [26]. Hogan et al. [29] therefore propose to let KGs return the most specific content type as possible. We measure the *Support of content negotiation* by dereferencing resources with different RDF serialization formats as desired content type and by comparing the accept header of the HTTP request with the content type of the HTTP response.

$$m_{Negot}(h_g) = \begin{cases} 1 & \text{CN supported and correct content types returned} \\ 0.5 & \text{CN supported but wrong content types returned} \\ 0 & \text{otherwise} \end{cases}$$

Linking HTML sites to RDF serializations Heath et al. [26] suggest linking any HTML description of a resource to RDF serializations of this resource in order to make the discovery of corresponding RDF data easier (for Linked Data aware applications). For that reason, in the HTML header the so-called *Autodiscov-*

[26] See `http://pingdom.com/`, requested on Mar 1, 2016.

ery pattern can be included. This pattern consists of the phrase `link rel=alternate`, the indication about the provided RDF content type, and a link to the RDF document.[27] We measure the linking of HTML pages to RDF documents (i.e., resource representations) by evaluating whether the HTML representations of the resources contain links as described:

$$m_{HTMLRDF}(h_g) = \begin{cases} 1 & \textit{Autodiscovery pattern} \text{ used} \\ & \text{at least once} \\ 0 & \text{otherwise} \end{cases}$$

Provisioning of KG metadata In the light of the Semantic Web vision where agents select and make use of appropriate data sources on the Web, also the meta-information about KGs needs to be available in a machine-readable format. The two important mechanisms to specify metadata about KGs are (i) using semantic sitemaps and (ii) using the VoID vocabulary[28] [26]. For instance, the URI of the SPARQL endpoint can be assigned via `void:sparqlEndpoint` and the RDF export URL can be specified with `void:dataDump`. Such metadata can be added as additional facts to the KG or it can be provided as separate VoID file. We measure the *Provisioning of KG metadata* by evaluating whether machine-readable metadata about the KG is available. Note that the provisioning of licensing information in a machine-readable format (which is also a meta-information about the KG) is considered in the data quality dimension *License* later on.

$$m_{Meta}(g) = \begin{cases} 1 & \text{Machine-readable metadata} \\ & \text{about } g \text{ available} \\ 0 & \text{otherwise} \end{cases}$$

3.5.2. License

Definition of dimension. Licensing is defined as "the granting of permission for a consumer to re-use a dataset under defined conditions" [49].

Discussion. The publication of licensing information about KGs is important for using KGs without legal concerns, especially in commercial settings. Creative Commons (CC)[29] publishes several standard licensing contracts which define rights and obligations. These contracts are also in the Linked Data context popular. The most frequent licenses for Linked Data are CC-BY, CC-BY-SA, and CC0 [31]. CC-BY[30] requires specifying the source of the data, CC-BY-SA[31] requires in addition that if the data is published, it is published under the same legal conditions; CC0[32] defines the respective data as public domain and without any restrictions.

Noteworthy is that most data sources in the Linked Open Data cloud do not provide any licensing information [31] which makes it difficult to use the data in commercial settings. Even if data is published under CC-BY or CC-BY-SA, the data is often not used since companies refer to uncertainties regarding these contracts.

Definition of metric. The dimension *License* is determined by the criterion *Provisioning machine-readable licensing information*.

The fulfillment degree of a KG g w.r.t. the dimension *License* is measured by the metric $m_{macLicense}$, which is defined as follows.

Provisioning machine-readable licensing information Licenses define the legal frameworks under which the KG data may be used. Providing machine-readable licensing information allows users and applications to be aware of the license and to use the data of the KG in accordance with the legal possibilities [30,26].

Licenses can be specified in RDF via relations such as `cc:licence`,[33] `dcterms:licence`, or `dcterms:rights`. The licensing information can be specified either in the KG as additional facts or separately in a VoID file. We measure the criterion by evaluating whether licensing information is available in a machine-readable format:

$$m_{macLicense}(g) = \begin{cases} 1 & \text{machine-readable} \\ & \text{licensing information} \\ & \text{available} \\ 0 & \text{otherwise} \end{cases}$$

3.5.3. Interlinking

Definition of dimension. Interlinking is the extent "to which entities that represent the same concept are

[27] An example is `<linkrel="alternate" type = "application/rdf+xml" href="company.rdf">`.

[28] See namespace `http://www.w3.org/TR/void`.

[29] See `http://creativecommons.org/`, requested on Mar 1, 2016.

[30] See `https://creativecommons.org/licenses/by/4.0/`, requestedon Mar 1, 2016.

[31] See `https://creativecommons.org/licenses/by-sa/4.0/`, requested on Mar 1, 2016.

[32] See `http://creativecommons.org/publicdomain/zero/1.0/`, requested on Mar 3, 2016.

[33] Using the namespace `http://creativecommons.org/ns#`.

linked to each other, be it within or between two or more data sources" [49].

Discussion. According to Bizer et al. [12], DBpedia established itself as a hub in the Linked Data cloud due to its intensive interlinking with other KGs. These interlinking is on the instance level usually established via `owl:sameAs` links. However, according to Halpin et al. [24], those `owl:sameAs` links do not always interlink identical entities in reality. According to the authors, one reason might be that the KGs provide entries in different granularity: For instance, the DBpedia resource for "Berlin" (`dbo:Berlin`) links via `owl:sameAs` relations to three different resources in the KG GeoNames,[34] namely (i) Berlin, the capital,[35] (ii) Berlin, the state,[36] and (iii) Berlin, the city.[37] Moreover, `owl:sameAs` relations are often created automatically by some mapping function. Due to mapping errors, the precision is often below 100% [18].

Definition of metric. The dimension *Interlinking* is determined by the criteria

- *Interlinking via `owl:sameAs`*
- *Validity of external URIs*

The fulfillment degree of a KG g w.r.t. the dimension *Interlinking* is measured by the metrics m_{Inst} and m_{URIs}, which are defined as follows.

Interlinking via `owl:sameAs` The forth Linked Data principle according to Berners-Lee [8] is the interlinking of data resources so that the user can explore further information. According to Hogan et al. [30], the interlinking has a side effect: It does not only result in otherwise isolated KGs, but the number of incoming links of a KG indicates the importance of the KG in the Linked Open Data cloud. We measure the interlinking on instance level[38] by calculating the extent to which instances have at least one `owl:sameAs` link to external KGs:

$$m_{Inst}(g) = |\{x \in I_g \setminus (P_g \cup C_g) \mid \exists(x, \texttt{owl:sameAs}, y) \in g \wedge y \in U_g^{ext}\}| \; / |I_g \setminus (P_g \cup C_g)|$$

Validity of external URIs The considered KG may contain outgoing links referring to RDF resources or Web documents (non-RDF data). The linking to RDF resources is usually enabled by `owl:sameAs`, `owl:equivalentProperty`, and `owl:equivalentClass` relations. Web documents are linked via relations such as `foaf:homepage` and `foaf:depiction`. Linking to external resources always entails the problem that those links might get invalid over time. This can have different causes. For instance, the URIs are not available anymore. We measure the *Validity of external URIs* by evaluating the URIs from an URI sample set w.r.t. whether there is a timeout, a client error (HTTP response 4xx) or a server error (HTTP response 5xx).

$$m_{URIs}(g) = \frac{|\{x \in A \mid resolvable(x)\}|}{|A|}$$

where $A = \{y \mid \exists(x, p, y) \in g : (p \in P_{eq} \wedge x \in U_g \setminus (C_g \cup P_g) \wedge x \in U_g^{local} \wedge y \in U_g^{ext})\}$ and $resolvable(x)$ returns true if HTTP status code 200 is returned. P_{eq} is the set of relations used for linking to external sources. Examples for such relations are `owl:sameAs` and `foaf:homepage`.

In case of an empty set A, the metric should evaluate to 1.

3.6. Conclusion

In this section, we provided 34 DQ criteria which can be applied in the form of DQ metrics to KGs in order to assess those KGs w.r.t. data quality. The DQ criteria are classified into 11 DQ dimensions. These dimensions are themselves grouped into 4 DQ categories. In total, we have the following picture:

- Intrinsic category
  - Accuracy
    - Syntactic validity of RDF documents
    - Syntactic validity of literals
    - Semantic validity of triples

[34] See `http://www.geonames.org/`, requested on Dec 31, 2016.

[35] See `http://www.geonames.org/2950159/berlin.html`, requested on Feb 4, 2017.

[36] See `http://www.geonames.org/2950157/land-berlin.html`, requested on Feb 4, 2017.

[37] See `http://www.geonames.org/6547383/berlin-stadt.html`, requested on Feb 4, 2017.

[38] The interlinking on schema level is already measured via the criterion *Interoperability of proprietary vocabulary*.

    - Trustworthiness
        - Trustworthiness on KG level
        - Trustworthiness on statement level
        - Using unknown and empty values
    - Consistency
        - Check of schema restrictions during insertion of new statements
        - Consistency of statements w.r.t. class constraints
        - Consistency of statements w.r.t. relation constraints
- Contextual category
    - Relevancy
        - Creating a ranking of statements
    - Completeness
        - Schema completeness
        - Column completeness
        - Population completeness
    - Timeliness
        - Timeliness frequency of the KG
        - Specification of the validity period of statements
        - Specification of the modification date of statements
- Representational data quality
    - Ease of understanding
        - Description of resources
        - Labels in multiple languages
        - Understandable RDF serialization
        - Self-describing URIs
    - Interoperability
        - Avoiding blank nodes and RDF reification
        - Provisioning of several serialization formats
        - Using external vocabulary
        - Interoperability of proprietary vocabulary
- Accessibility category
    - Accessibility
        - Dereferencing possibility of resources
        - Availability of the KG
        - Provisioning of public SPARQL endpoint
        - Provisioning of an RDF export
        - Support of content negotiation
        - Linking HTML sites to RDF serializations
        - Provisioning of KG metadata
    - License
        - Provisioning machine-readable licensing information
    - Interlinking
        - Interlinking via `owl:sameAs`
        - Validity of external URIs

4. Selection of KGs

We consider the following KGs for our comparative evaluation:

- **DBpedia**: DBpedia[39] is the most prominent KG in the LOD cloud [4]. The project was initiated by researchers from the Free University of Berlin and the University of Leipzig, in collaboration with OpenLink Software. Since the first public release in 2007, DBpedia is updated roughly once a year.[40] By means of a dedicated open source extraction framework, DBpedia is created from information contained in Wikipedia, such as infobox tables, categorization information, geo-coordinates, and external links. Due to its role as the hub of the LOD cloud, DBpedia contains many links to other datasets in the LOD cloud such as Freebase, OpenCyc, UMBEL,[41] GeoNames, Musicbrainz,[42] CIA World Factbook,[43] DBLP,[44] Project Gutenberg,[45] DBtune Jamendo,[46] Eurostat,[47] Uniprot,[48] and Bio2RDF.[49,50] DBpedia has been used extensively in the Semantic Web research community, but has become also relevant in commercial settings: for instance, companies such as the BBC [33] and the New York Times [41] use DBpedia to organize their content. The version of DBpedia we analyzed is 2015-04.

[39] See `http://dbpedia.org`, requested on Nov 1, 2016.

[40] There is also *DBpedia live* which started in 2009 and which gets updated when Wikipedia is updated. See `http://live.dbpedia.org/`, requested on Nov 1, 2016. Note, however, that DBpedia live only provides a restricted set of relations compared to DBpedia. Also, the provisioning of data varies a lot: While for some time ranges DBpedia live provides data for each hour, for other time ranges DBpedia live data is only available once a month.

[41] See `http://umbel.org/`, requested on Dec 31, 2016.

[42] See `http://musicbrainz.org/`, requested on Dec 31, 2016.

[43] See `https://www.cia.gov/library/publications/the-world-factbook/`, requested on Dec 31, 2016.

[44] See `http://www.dblp.org`, requested on Dec 31, 2016.

[45] See `https://www.gutenberg.org/`, requested on Dec 31, 2016.

[46] See `http://dbtune.org/jamendo/`, requested on Dec 31, 2016.

[47] See `http://eurostat.linked-statistics.org/`, requested on Dec 31, 2016.

[48] See `http://www.uniprot.org/`, requested on Dec 31, 2016.

[49] See `http://bio2rdf.org/`, requested on Dec 31, 2016.

[50] See a complete list of the links on the websites describing the single DBpedia versions such as `http://downloads.dbpedia.org/2016-04/links/` (requested on Nov 1, 2016).

– **Freebase**: Freebase[51] is a KG announced by Metaweb Technologies, Inc. in 2007 and was acquired by Google Inc. on July 16, 2010. In contrast to DBpedia, Freebase had provided an interface that allowed end-users to contribute to the KG by editing structured data. Besides user-contributed data, Freebase integrated data from Wikipedia, NNDB,[52] FMD,[53] and MusicBrainz.[54] Freebase uses a proprietary graph model for storing also complex statements. Freebase shut down its services completely on August 31, 2016. Only the latest data dump is still available. Wikimedia Deutschland and Google integrate Freebase data into Wikidata via the *Primary Sources Tool*.[55] Further information about the migration from Freebase to Wikidata is provided in [44]. We analyzed the latest Freebase version as of March 2015.
– **OpenCyc**: The Cyc[56] project started in 1984 by the industry research and development consortium Microelectronics and Computer Technology Corporation. The aim of Cyc is to store – in a machine-processable way – millions of common sense facts such as "Every tree is a plant." The main focus of Cyc has been on inferencing and reasoning. Since Cyc is proprietary, a smaller version of the KG called OpenCyc[57] was released under the open source Apache license Version 2. In July 2006, ResearchCyc[58] was published for the research community, containing more facts than OpenCyc. We did not consider Cyc and ResearchCyc, since those KGs do not meet the chosen requirements, namely, that the KGs are freely available and freely usable in any context. The version of OpenCyc we analyzed is 2012-05-10.
– **Wikidata**: Wikidata[59] is a project of Wikimedia Deutschland which started on October 30, 2012. The aim of the project is to provide data which can be used by any Wikimedia project, including Wikipedia. Wikidata does not only store facts, but also the corresponding sources, so that the validity of facts can be checked. Labels, aliases, and descriptions of entities in Wikidata are provided in almost 400 languages. Wikidata is a community effort, i.e., users collaboratively add and edit information. Also, the schema is maintained and extended based on community agreements. Wikidata is currently growing considerably due to the integration of Freebase data [44]. The version of Wikidata we analyzed is 2015-10.
– **YAGO**: YAGO[60] – Yet Another Great Ontology – has been developed at the Max Planck Institute for Computer Science in Saarbrücken since 2007. YAGO comprises information extracted from Wikipedia (such as information from the categories, redirects, and infoboxes), WordNet [19] (such as information about synsets and hyponomies), and GeoNames.[61] The version of YAGO we analyzed is YAGO3, which was published in March 2015.

5. Comparison of KGs

5.1. Key Statistics

In the following, we present statistical commonalities and differences of the KGs DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. We thereby use the following key statistics:

– Number of triples
– Number of classes
– Number of relations
– Distribution of classes w.r.t. the number of their corresponding instances
– Coverage of classes with at least one instance per class
– Covered domains w.r.t. entities
– Number of entities
– Number of instances
– Number of entities per class
– Number of unique subjects
– Number of unique predicates
– Number of unique objects

In Section 7.2, we provide an overview of related work w.r.t. those key statistics.

[51] See http://freebase.com/, requested on Nov 1, 2016.
[52] See http://www.nndb.com, requested on Dec 31, 2016.
[53] See http://www.fashionmodeldirectory.com/, requested on Dec 31, 2016.
[54] See http://musicbrainz.org/, requested on Dec 31, 2016.
[55] See https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool, requested on Apr 8, 2016.
[56] See http://www.cyc.com/, requested on Dec 31, 2016.
[57] See http://www.opencyc.org/, accessed on Nov 1, 2016.
[58] See http://research.cyc.com/, requested on Dec 31, 2016.
[59] See http://wikidata.org/, accessed on Nov 1, 2016.
[60] See http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/, accessed on Nov 1, 2016.
[61] See http://www.geonames.org/, requested on Dec 31, 2016.

5.1.1. Triples

Ranking of KGs w.r.t. number of triples. The number of triples (see Table 2) differs considerably between the KGs: Freebase is the largest KG with over 3.1B triples, while OpenCyc resides the smallest KG with only 2.4M triples. The large size of Freebase can be traced back to the fact that large data sets such as MusicBrainz have been integrated into this KG. OpenCyc, in contrast, has been built purely manually by experts. In general, this indicates a correlation between the way of building up a KG and its size.

Size differences between DBpedia and YAGO. As both DBpedia and YAGO were created automatically by extracting semantically-structured information from Wikipedia, the significant difference between their sizes – in terms of triples – is in particular noteworthy. We can mention here the following reasons: YAGO integrates the statements from different language versions of Wikipedia in one single KG while for the canonical DBpedia dataset (which is used in our evaluations) solely the English Wikipedia was used as information source. Besides that, YAGO contains contextual information and detailed provenance information. Contextual information is for instance the anchor texts of all links within Wikipedia. For representing the anchor texts, the relation `yago:hasWikipediaAnchorText` (330M triples in total) is used. The provenance information of single statements is stored in a reified form. In particular, the relations `yago:extractionSource` (161.2M triples) and `yago:extractionTechnique` (176.2M triples) are applied therefore.

$3n$

Influence of reification on the number of triples. DBpedia, Freebase, Wikidata, and YAGO use some form of reification. Reification in general describes the possibility of making statements about statements. While reification has an influence on the number of triples for DBpedia, Freebase, and Wikidata, the number of triples in YAGO is not influenced by reification since data is here provided in N-Quads.[62] This style of reification is called *Named Graph* [27]: The additional column (in comparison to triples) contains a unique ID of the statement by which the triple becomes identified. For backward compatibility the ID is commented and therefore not imported into the triple store. Note, however, that transforming N-Quads to N-Triples leads to a high number of unique subjects concerning the set of all triples.

In case of DBpedia, Freebase, and Wikidata, reification is implemented by means of *n-ary relations*. An n-ary relation denotes the relation between more than two resources and is implemented via additional, intermediate nodes, since in RDF only binary statements can be modeled [16,27]. In Freebase and DBpedia, data is mostly provided in the form of plain N-Triples and n-ary relations are only used for data from higher arity.[63] Wikidata, in contrast, has the peculiarity that not only every statement is expressed with the help of an n-ary relation, but that in addition each statement is instantiated with `wdo:Statement`. This leads to about 74M additional instances, which is about one tenth of all triples in Wikidata.

5.1.2. Classes

Methods for counting classes. The number of classes can be calculated in different ways: Classes can be identified via `rdfs:Class` and `owl:Class` relations, or via `rdfs:subClassOf` relations.[64] Since Freebase does not provide any class hierarchy with `rdfs:subClassOf` relations and since Wikidata does not instantiate classes explicitly as classes, but uses instead only "subclass of" (`wdt:P279`) relations, the method of calculating the number of classes depends on the considered KG.

Ranking of KG w.r.t. number of classes. Our evaluations revealed that YAGO contains the highest number of classes of all considered KGs; DBpedia, in contrast, has the fewest (see Table 2).

Number of classes in YAGO and DBpedia. How does it come to this gap between DBpedia and YAGO with respect to the number of classes, although both KGs were created automatically based on Wikipedia? For YAGO, the classes are extracted from the categories in Wikipedia, while the hierarchy of the classes is deployed with the help of WordNet synset relations. The DBpedia ontology, in contrast, is very small, since it is created manually, based on the mostly used infobox

[62] The idea of N-Quads is based on the assignment of triples to different graphs. YAGO uses N-Quads to identify statements per ID.

[63] In Freebase *Compound Value Types* are used for reification [44]. In DBpedia it is named *Intermedia Node Mapping*, see `http://mappings.dbpedia.org/index.php/Template:IntermediateNodeMapping` (requested on Dec 31, 2016).

[64] The number of classes in a KG may also be calculated by taking all entity type relations (`rdf:type` and "instance of" (`wdt:P31`) in case of Wikidata) on the instance level into account. However, this would result only in a lower bound estimation, as here those classes are not considered which have no instances.

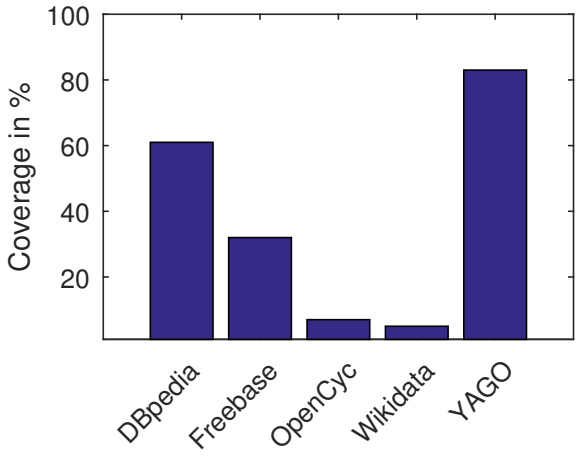

Fig. 1. Coverage of classes having at least one instance.

templates in Wikipedia. Besides those 736 classes, the DBpedia KG contains further 444,895 classes which originate from the imported YAGO classes and which are published in the namespace `yago:`. Those YAGO classes are – like the DBpedia ontology classes – interconnected via `rdfs:subClassOf` to form a taxonomy. In the evaluation of DBpedia, the YAGO classes are ignored, as they do not belong to the DBpedia ontology given as OWL file.

Coverage of classes with at least one instance. Fig. 1 shows for each KG the extent to which classes are instantiated, that is, for how many classes at least one instance exists. YAGO exhibits the highest coverage rate (82.6%), although it contains the highest number of classes among the KGs. This can be traced back to the fact that YAGO classes are chosen by a heuristic that considers Wikipedia *leaf categories* which tend to have instances [43]. OpenCyc (with 6.5%) and Wikidata (5.4%) come last in the ranking. Wikidata has the second highest number of classes in total (see Table 2), out of which relatively little are used on instance level. Note, however, that in some scenarios solely the schema level information (including classes) of KGs is necessary, so that the low coverage of instances by classes is not necessarily an issue.

Correlation between number of classes and number of instances. In Fig. 2, we can see a histogram of the classes with respect to the number of instances per class. That is, for each KG we can spot how many classes have a high number of instances and how many classes have a low number of instances. Note the logarithmic scale on both axes. The curves seem to follow power law distributions. For DBpedia, the line decreases consistently for the first 250 classes, before it decreases more than exponentially beyond class 250.

Table 1
Percentage of considered entities per KG for covered domains

	DB	FB	OC	WD	YA
Reach of method	88%	92%	81%	41%	82%

5.1.3. Domains

All considered KGs are cross-domain, meaning that a variety of domains are covered in those KGs. However, the KGs often cover the single domains to a different degree. Tartir [45] proposed to measure the covered domains of ontologies by determining the usage degree of corresponding classes: the number of instances belonging to one or more subclasses of the respective domain is compared to the number of all instances. In our work, however, we decided to evaluate the coverage of domains concerning the classes per KG via manual assignments of the mostly used classes to the domains *people*, *media*, *organizations*, *geography*, and *biology*.[65] This list of domains was created by aggregating the most frequent domains in Freebase.

The manual assignment of classes to domains is necessary in order to obtain a consistent assignment of the classes to the domains across all considered KGs. Otherwise, the same classes in different KGs may be assigned to different domains. Moreover, in some KGs classes may otherwise appear in various domains simultaneously. For instance, the Freebase classes `freebase:music.artist` and `freebase:people.person` overlap in terms of their instances and multiple domains (such as *music* and *people*) might be assigned to them.

As the reader can see in Table 1, our method to determine the coverage of domains, and, hence, the reach of our evaluation, includes about 80% of all entities of each KG, except Wikidata. It is calculated as the ratio of the number of unique entities of all considered domains of a given KG divided by the number of all entities of this KG.[66] If the ratio was at 100% we were able to assign all entities of a KG to the chosen domains.

Fig. 3 shows the number of entities per domain in the different KGs with a logarithmic scale. Fig. 4 presents

[65] See our website for examples of classes per domain and per KG `http://km.aifb.kit.edu/sites/knowledge-graph-comparison/` (requested on Dec 31, 2016).

[66] We used the number of unique entities of all domains and not the sum of the entities measured per domain, since entities may be in several domains at the same time.

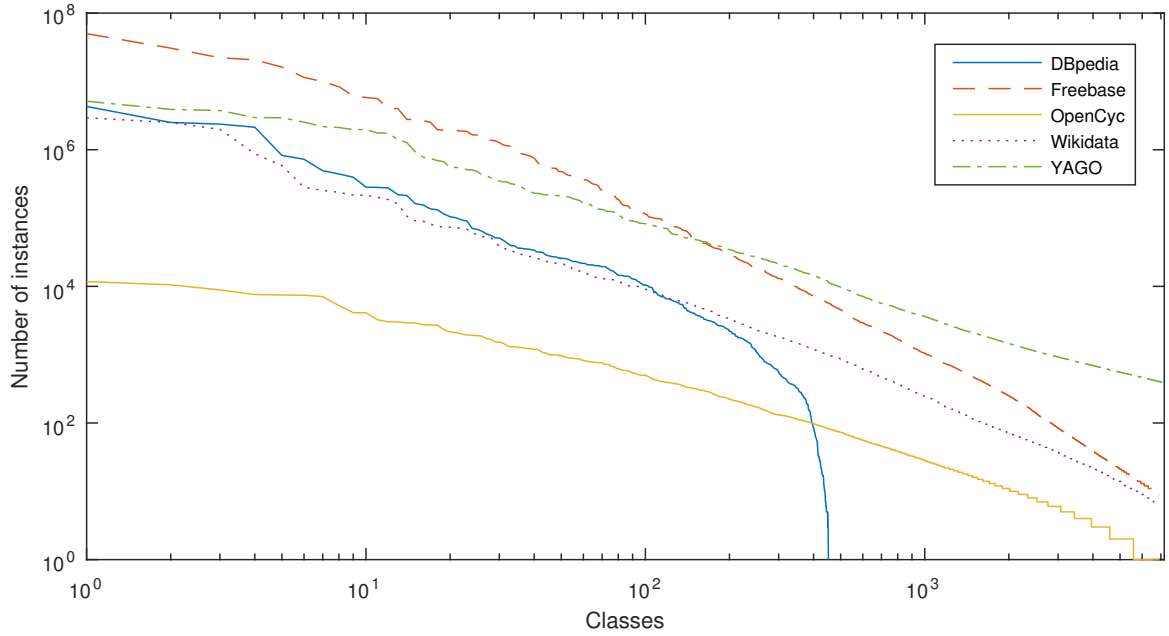

Fig. 2. Distribution of classes w.r.t. the number of instances per KG.

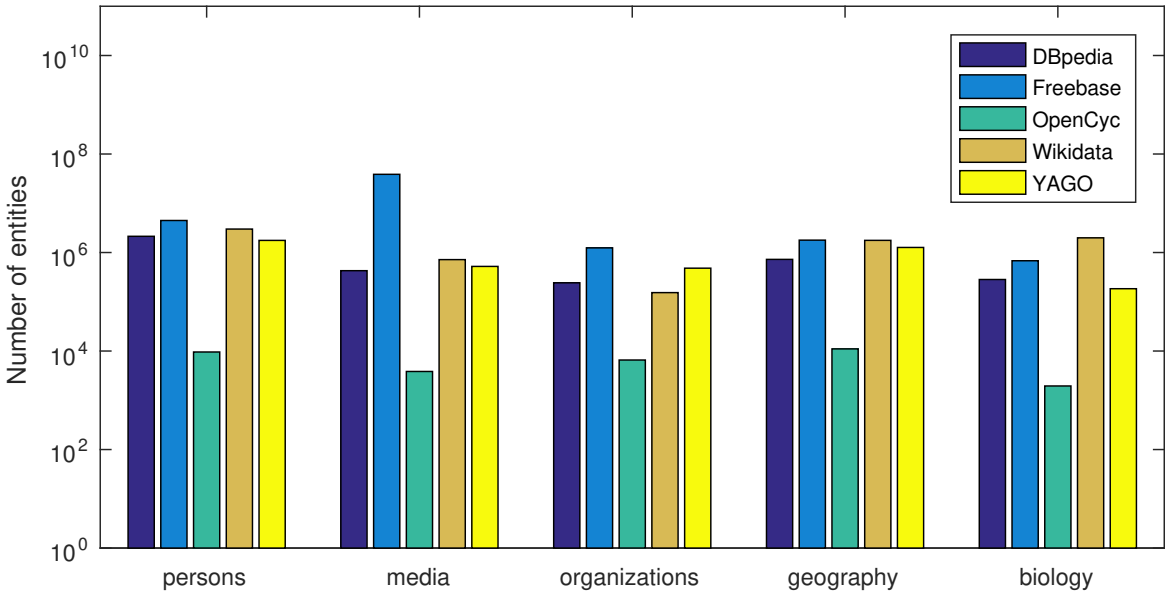

Fig. 3. Number of entities per domain.

the relative coverage of each domain in each KG. It is calculated as the ratio of the number of entities in each domain to the total number of entities of the KG. A value of 100% means that all instances reside in one single domain.

The case of Freebase is especially outstanding here: 77% of all entities here are located in the *media* domain. This fact can be traced back to large-scale data imports, such as from MusicBrainz. The class `freebase:music.release_track` is accountable for 42% of the media entities. As shown in Fig. 3, Freebase provides the most entities in four out of the five domains when considering all KGs.

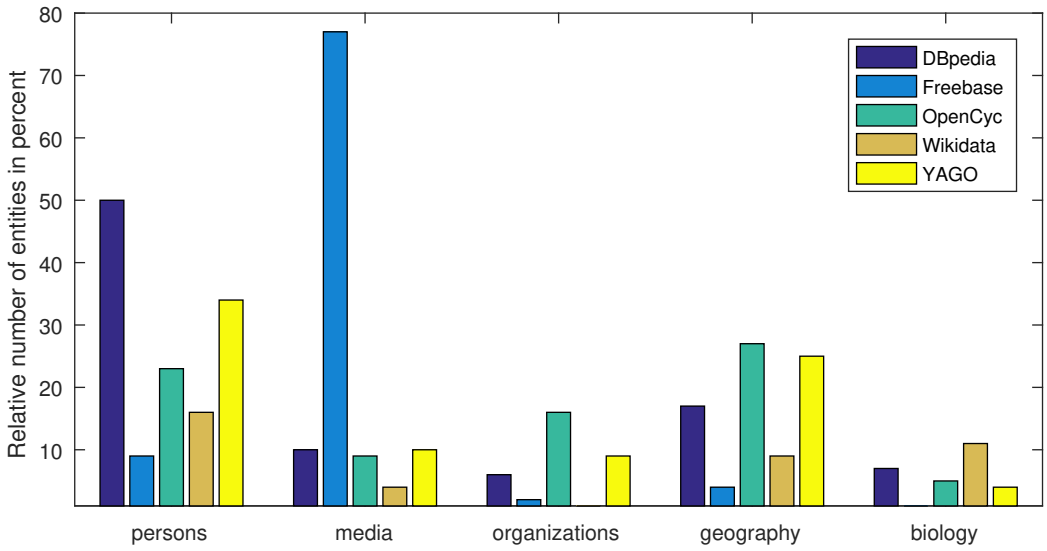

Fig. 4. Relative number of entities per domain.

In DBpedia and YAGO, the domain of *people* is the largest domain (50% and 34%, respectively). Peculiar is the higher coverage of YAGO regarding the *geography* domain compared to DBpedia. As one reason for that we can point out the data import of GeoNames into YAGO.

Wikidata contains around 150K entities in the domain *organization.* This is relativly few considering the total amount of entities being around 18.7M and considering the number of organizations in other KGs. Note that even DBpedia provides more organization entities than Wikidata. The reason why Wikidata has not so many organization entities is not fully comprehensible to us. However, we can point out that for our analysis we only considered Wikidata classes which appeared more than 6,000 times[67] and that about 16K classes were therefore not considered. It is possible that entities of the domain *organization* are belonging to those rather rarely occurring classes.

5.1.4. Relations and Predicates

Evaluation method. In this article, we differentiate between *relations* and *predicates* (see also Section 2):

- *Relations* – as short term for explicitly defined relations – refers to (proprietary) vocabulary defined on the schema level of a KG. We identify the set of relations of a KG as the set of those links which are explicitly defined as such via assignments (for instance, with `rdfs:Property`) to classes. In Section 2 we used P_g to denote this set.
- In contrast, we use *predicates* to denote links used in the KG independently of their introduction on the schema level. The set of unique predicates per KG, denoted as P_g^{imp}, is nothing else than the set of unique RDF terms on the predicate position of all triples in the KG.

It is important to distinguish the key statistics for relations from the key statistics for predicates, since they can differ considerably, depending on to which degree relations are only defined on schema level, but not used on instance level.

Evaluation results.

Relations

Ranking regarding relations. As presented in Table 2, Freebase exhibits by far the highest number of unique relations (around 785K) among the KGs. YAGO shows only 106 relations, which is the lowest value in this comparison. In the following, we point out further findings regarding the relations of the single KGs.

DBpedia Regarding DBpedia relations we need to distinguish between so-called *mapping-based properties* and *non-mapping-based properties*. Mapping-based properties are created by extracing the information from infoboxes in Wikipedia using manually created mappings. These mappings are specified in the DB-

[67] This number is based on heuristics. We focused on the 150 most instantiated classes and cut the long tail of classes having only few instances.

pedia Mappings Wiki.[68] Mapping-based properties are contained in the DBpedia ontology and located in the namespace `http://dbpedia.org/ontology/`. We count 2,819 such relations for the considered DBpedia version 2015-04. Non-mapping-based properties (also called "raw infobox properties") are extracted from Wikipedia without the help of manually created mappings and, hence, without any manual adjustments. Therefore, they are generally of lower quality. We count 58,776 such unique relations. They reside in the namespace `http://dbpedia.org/property/`. Both mapping-based and non-mapping-based properties are instantiated in DBpedia with `rdf:Property`. We ignore the non-mapping based properties for the calculation of the number of relations, $|P_g|$, (see Table 2), since, in contrast to DBpedia, in YAGO non-mapping based properties are not instantiated. Note that the mapping-based properties and the non-mapping based properties in DBpedia are not aligned[69] and may overlap until DBpedia version 2016-04.[70]

Freebase The high number or Freebase relations can be explained by two facts: 1. About a third of all relations in Freebase are duplicates in the sense that they are declared by means of the `owl:inverseOf` relation as being inverse of other relations. An example is the relation `freebase:music.artist.album` and its inverse relation `freebase:music.album.artist`. 2. Freebase allowed users to introduce their own relations without any limits. These relations were originally in each user's namespace. So-called *commons admins* were able to approve those relations so that they got included into the Freebase commons schema.

OpenCyc For OpenCyc we measure 18,028 unique relations. We can assume that most of them are dedicated to statements on the schema level.

Wikidata In Wikidata a relatively small set of relations is provided. Note in this context that, despite the fact that Wikidata is curated by a community (just like Freebase), Wikidata community members cannot insert arbitrarily new relations as it was possible in Freebase; instead, relations first need to be proposed and then get accepted by the community if and only if certain criteria are met.[71] One of those criteria is that each new relation is presumably used at least 100 times. This relation proposal process can be mentioned as likely reason why in Wikidata in relative terms more relations are actually used than in Freebase.

YAGO For YAGO we measure the small set of 106 unique relations. Although relations are curated manually for YAGO and DBpedia, the size of the relation set differs significantly between those KGs. Hoffart et al. [28] mention the following reasons for that:

1. *Peculiarity of relations:* The DBpedia ontology provides quite many special relations. For instance, there exists the relation `dbo:aircraftFighter` between `dbo:MilitaryUnit` and `dbo:MeanOfTransportation`.
2. *Granularity of relations:* Relations in the DBpedia ontology are more fine-grained than relations in YAGO. For instance, DBpedia contains the relations `dbo:author` and `dbo:director`, whereas in YAGO there is only the generic relation `yago:created`.
3. *Date specification:* The DBpedia ontology introduces several relations for dates. For instance, DBpedia contains the relations `dbo:birthDate` and `dbo:birthYear` for birth dates, while in YAGO only the relation `yago:birthOnDate` is used. Incomplete date specifications – for instance, if only the year is known – are specified in YAGO by wildcards ("#"), so that no multiple relations are needed.
4. *Inverse relations:* YAGO has no relations explicitly specified as being inverse. In DBpedia, we can find relations specified as inverse such as `dbo:parent` and `dbo:child`.
5. *Reification:* YAGO introduces the SPOTL(X) format. This format extends the triple format "SPO" with a specification of Time, Location and conteXt. In this way, no contextual relations are necessary (such as `dbo:distanceToLondon` or `dbo:populationAsOf`), which occur if the relations are closely aligned to Wikipedia template attribute names.

Frequency of the usage of relations. Fig. 5 shows the relative proportions of how often relations are used per KG, grouped into three classes. Surprisingly, DBpedia and Freebase exhibit a high number of relations which are not used at all on the instance level. In case of

[68] See `http://mappings.dbpedia.org/index.php/Main_Page`, accessed on Nov 4, 2016.

[69] For instance, The DBpedia ontology contains `dbo:birthName` for the name of a person, while the non-mapping based property set contains `dbp:name`, `dbp:firstname`, and `dbp:alternativeNames`.

[70] For instance, `dbp:alias` and `dbo:alias`.

[71] See `https://www.wikidata.org/wiki/Wikidata:Property_proposal`, requested on Dec 31, 2016.

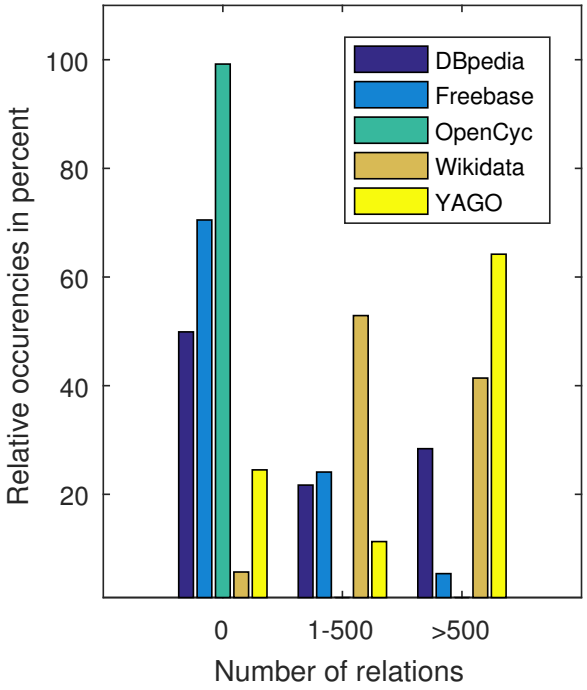

Fig. 5. Frequency of the usage of the relations per KG, grouped by (i) zero occurrences, (ii) 1–500 occurrences, and (iii) more than 500 occurrences in the respective KG.

OpenCyc, 99.2% of the defined relations are never used. We assume that those relations are used only within Cyc, the commercial version of OpenCyc. In case of Freebase, only 5% of the relations are used more than 500 times and about 70% are not used at all. Analogously to the discussion regarding the number of Freebase relations, we can mention again the high number of defined `owl:inverseOf` relations and the high number of users' relation proposals as reasons for that.

Predicates

Ranking regarding predicates. Freebase is here – like in case of the ranking regarding relations – ranked first. The lowest number of unique predictes is provided by OpenCyc, which exhibits only 165 predicates. All KGs except OpenCyc provide more predicates then relations. Our single observations regarding the predicate sets are as follows:

DBpedia DBpedia is ranked third in terms of the absolute numbers of predicates: about 60K predicates are used in DBpedia. The set of relations and the set of predicates varies considerably here, since also facts are extracted from Wikipedia info-boxes whose predicates are considered by us as being only implicitly defined and which, hence, occur only as predicates. These are the so-called non-mapping-based properties. Note that in the studied DBpedia version 2015-04 the set of explicitly defined relations (mapping-based properties) and the set of implicitly defined relations (non-mapping-based properties) overlaps. An example is `dbp:alias` with `dbo:alias`.

Freebase We can observe here a similar picture as for the set of Freebase relations: With about 785K unique predicates, Freebase exceeds the other KGs by far. Note, however, that 95% of the predicates (around 743K) are used only once. This relativizes the high number. Most of the predicates are keys in the sense of ids and are used for internal modeling (for instance, `freebase:key.user.adrianb`).

OpenCyc In contrast to the 18,028 unique relations, we measure only 164 unique predicates for OpenCyc. More predicates are presumably used in Cyc.

Wikidata We measure more Wikidata predicates than Wikidata relations, since Wikidata predicates are created by modifying Wikidata relations. An example are the following triples, which express the statement "Barack Obama (`wdt:Q76`) is a human (`wdt:Q5`)" by an intermediate node (`wdt:Q76S123`, abbreviated):

```
wdt:Q76 wdt:P31s wdt:Q76S123.
wdt:Q76S123 wdt:P31v wdt:Q5.
```

The relation extension "s" indicates that the RDF term in the object position is a statement. The "v" extension allows to refer to a value (in Wikidata terminology). Besides those extensions, there is "r" to refer to a reference and the "q" extension to refer to a qualifier. In general, these relation extensions are used for realizing reification via n-ary relations. For that, intermediate nodes are used which represent statements [16].

YAGO YAGO contains more predicates than DBpedia, since infobox attributes from different language versions of Wikipedia are aggregated into one KG,[72] while for DBpedia separate, localized KG versions are offered for non-English languages.

5.1.5. Instances and Entities

Evaluation method. We distinguish between instances I_g and entities E_g of a KG (cf. Section 2).

1. *Instances* are belonging to classes. They are identified by retrieving the subjects of all triples where the predicates indicate class affiliations.

[72] The language of each attribute is encoded in the URI, for instance `yago:infobox/de/fläche` and `yago:infobox/en/areakm`.

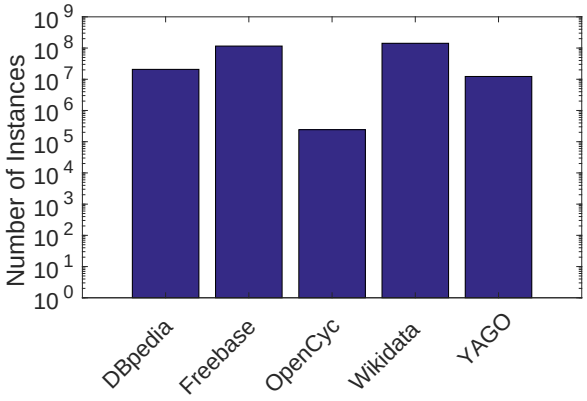

Fig. 6. Number of instances per KG.

2. *Entities* are real-world objects. This excludes, for instance, instantiated statements for being entities. Determining the set of entities is partially tricky: In DBpedia and YAGO entities are determined as being an instance of the class `owl:Thing`. In Freebase entities are instances of `freebase:common.topic` and in Wikidata instance of `wdo:Item`. In OpenCyc, `cych:Individual` corresponds to `owl:Thing`, but not all entities are classified in this way. Therefore, we approximately determine the set of entities in OpenCyc by manually classifying all classes having more than 300 instances, including at least one entity.[73] In this way, abstract classes such as `cych:ExistingObjectType` are neglected.

Ranking w.r.t. the number of instances. Table 2 and Fig. 6 show the number of instances per KG. We can see that Wikidata comprises the highest number of instances (142M) in total and OpenCyc the fewest (242K).

Ranking w.r.t. the number of entities. Table 2 shows the ranking of KGs regarding the number of entities. Freebase contains by far the highest number of entities (about 49.9M). OpenCyc is at the bottom with only about 41K entities.

Differences in number of entities. The reason why the KGs show quite varying numbers of entities are the information sources of the KGs. We illustrate this with the music domain as example:

1. *Freebase* had been created mainly from data imports such as from MusicBrainz. Therefore, entities in the domain of media and especially song release tracks are covered very well in Freebase: 77% of all entities are in the media domain (see Section 5.1.3), out of which 42% are release tracks.[74]
   Due to the large size and the world-wide coverage of entities in MusicBrainz, Freebase contains albums and release tracks of both English and non-English languages. For instance, regarding the English language, the album "Thriller" from Michael Jackson and its single "Billie Jean" are there, as well as rather unknown songs from the "Thriller" album such as "The Lady in My Life". Regarding non-English languages, Freebase contains for instance songs and albums from Helene Fischer such as "Lass' mich in dein Leben" and "Zaubermond;" also rather unknown songs such as "Hab' den Himmel berührt" can be found.
2. In case of *DBpedia*, the English Wikipedia is the source of information. In the English Wikipedia, many albums and singles of English artists are covered – such as the album "Thriller" and the single "Billie Jean." Rather unknown songs such as "The Lady in My Life" are not covered in Wikipedia. For many non-English artists such as the German singer Helene Fischer no music albums and no singles are contained in the English Wikipedia. In the corresponding language version of Wikipedia (and localized DBpedia version), this information is often available (for instance, the album "Zaubermond" and the song "Lass' mich in dein Leben"), but not the rather unknown songs such as "Hab' den Himmel berührt."
3. For *YAGO*, the same situation as for DBpedia holds, with the difference that YAGO in addition imports entities also from the different language versions of Wikipedia and imports also data from sources such as GeoNames. However, the above mentioned works ("Lass' mich in dein Leben," "Zaubermond," and "Hab' den Himmel berührt") of Helene Fischer are not in the YAGO, although the song "Lass' mich in dein Leben" exists in the German Wikipedia since May 2014 and although the used YAGO version 3 is based on the Wikipedia dump of June 2014.[75] Presumably, the YAGO extraction system was unable to extract any

[73] For instance, `cych:Individual`, `cych:Movie_CW` and `cych:City`.

[74] Those release tracks are expressed via `freebase:music.release_track`.

[75] See `http://www.mpi-inf.mpg.de/de/departments/databases-and-information-`

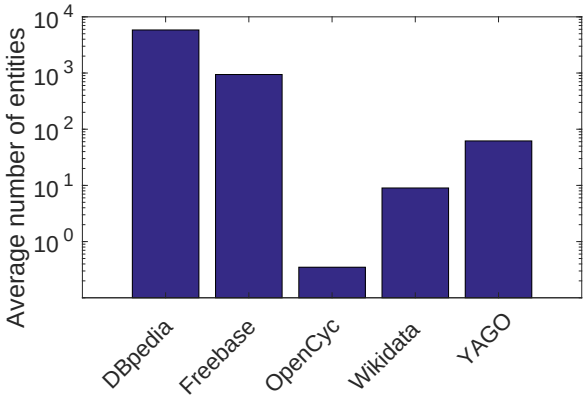

Fig. 7. Average number of entities per class per KG.

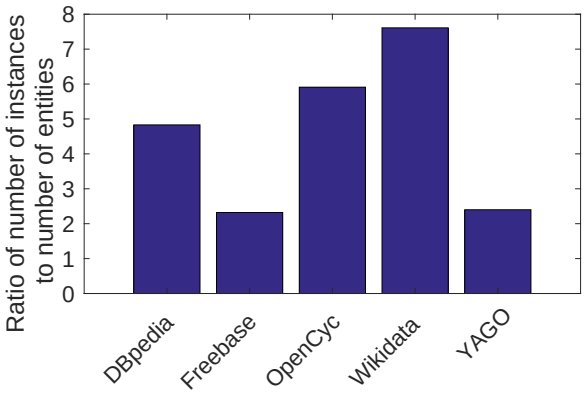

Fig. 8. Ratio of the number of instances to the number of entities for each KG.

types for those entities, so that those entities were discarded.

4. *Wikidata* is supported by the community and contains music albums of English and non-English artists, even if they do not exist in Wikipedia. An example is the song "The Lady in My Life." Note, however, that Wikidata does not provide all artist's works such as from Helene Fischer.
5. *OpenCyc* contains only very few entities in the music domain. The reason is that OpenCyc has its focus mainly on common-sense knowledge and not so much on facts about entities.

Average number of entities per class. Fig. 7 shows the average number of entities per class, which can be written as $|E_g|/|C_g|$. Obvious is the difference between DBpedia and YAGO (despite the similar number of entities): The reason for that is that the number of classes in the DBpedia ontology is small (as created manually) and in YAGO large (as created automatically).

Comparing number of instances with number of entities. Comparing the ratio of the number of instances to the number of entities for each KG, Wikidata exposes the highest difference. As reason for that we can state that each statement in Wikidata is modeled as an instance of `wdo:Statement`, leading to 74M additional instances. In other KGs such as DBpedia, statements are modeled without any dedicated statement assignment. OpenCyc exposes also a high ratio, since it contains mainly common sense knowledge and not as many entities as the other KGs. Furthermore, for our analysis we do not regard 100% of the entities, but only a large fraction of it (more precisely, the classes with the most frequently occurring instantiations), since entities are not consistently instantiated in OpenCyc (see beginning of Section 5.1.5).

5.1.6. Subjects and Objects

Evaluation method. The number of unique subjects and unique objects can be a meaningful KG characteristic regarding the link structure within the KG and in comparison to other KGs. Especially interesting are differences between the number of unique subjects and the number of unique objects.

We measure the number of unique subjects by counting the unique resources (i.e., URIs and blank nodes) on the subject position of N-Triples: $S_g := \{s \mid (s,p,o) \in g\}$. Furthermore, we measure the number of unique objects by counting the unique resources on the object position of N-Triples, excluding literals: $O_g := \{o \mid (s,p,o) \in g \wedge o \in U \cup B\}$. Complementary, the number of literals is given as: $O_g^{lit} := \{o \mid (s,p,o) \in g \wedge o \in L\}$.

Ranking of KGs regarding number of unique subjects. The number of unique subjects per KG is presented in Fig. 9. YAGO contains the highest number of different subjects, while OpenCyc contains the fewest.

Ranking of KGs regarding number of unique objects. The number of unique objects is also presented in Fig. 9. Freebase shows the highest score in this regard, OpenCyc again the lowest.

Ranking of KGs regarding the ratio of number of unique subjects to number of unique objects. The ratios of the number of unique subjects to the number of unique objects vary considerably between the KGs (see Fig. 8). We can observe that DBpedia has 2.65 times more objects than subjects, while YAGO on the other side has 19 times more unique subjects than objects.

`systems/research/yago-naga/yago/archive/`, requested on Dec 31, 2016.

Table 2
Summary of key statistics.

	DBpedia	Freebase	OpenCyc	Wikidata	YAGO
Number of triples $\|(s,p,o) \in g\|$	411 885 960	3 124 791 156	2 412 520	748 530 833	1 001 461 792
Number of classes $\|C_g\|$	736	53 092	116 822	302 280	569 751
Number of relations $\|P_g\|$	2819	70 902	18 028	1874	106
No. of unique predicates $\|P_g^{imp}\|$	60 231	784 977	165	4839	88 736
Number of entities $\|E_g\|$	4 298 433	49 947 799	41 029	18 697 897	5 130 031
Number of instances $\|I_g\|$	20 764 283	115 880 761	242 383	142 213 806	12 291 250
Avg. number of entities per class $\frac{\|E_g\|}{\|C_g\|}$	5840.3	940.8	0.35	61.9	9.0
No. of unique subjects $\|S_g\|$	31 391 413	125 144 313	261 097	142 278 154	331 806 927
No. of unique non-literals in obj. pos. $\|O_g\|$	83 284 634	189 466 866	423 432	101 745 685	17 438 196
No. of unique literals in obj. pos. $\|O_g^{lit}\|$	161 398 382	1 782 723 759	1 081 818	308 144 682	682 313 508

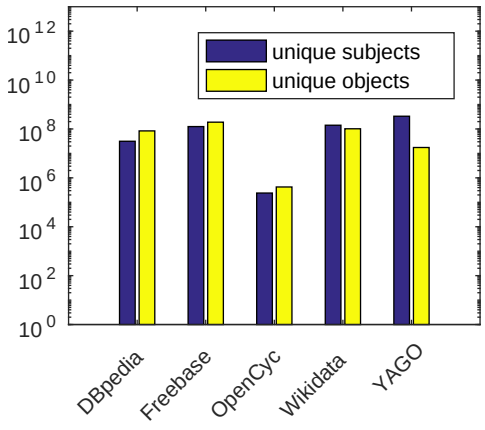

Fig. 9. Number of unique subjects and objects per KG. Note the logarithmic scale on the axis of ordinates.

The high number of unique subjects in YAGO is surprising and can be explained by the reification style used in YAGO. Facts are stored as N-Quads in order to allow for making statements about statements (for instance, storing the provenance information for statements). To that end, IDs (instead of blank nodes) which identify the triples are used on the first position of N-Triples. They lead to 308M unique subjects, such as `yago:id_6jg5ow_115_lm6jdp`. In the RDF export of YAGO, the IDs which identify the triples are commented out in order to facilitate the N-Triple format. However, the statements about statements are also transformed to triples. In those cases, the IDs identifying the reified statements are in the subject position, leading to such a high number of unique subjects.

DBpedia contains considerably more `owl:sameAs` links to external resources than KGs like YAGO (29.0M vs. 3.8M links), leading to a bias of DBpedia towards a high number of unique objects.

5.1.7. Summary of Key Statistics

Based on the evaluation results presented in the last subsections, we can highlight the following insights:

1. *Triples*: All KGs are very large. Freebase is the largest KG in terms of number of triples, while OpenCyc is the smallest KG. We notice a correlation between the way of building up a KG and the size of the KG: automatically created KGs are typically larger, as the burdens of integrating new knowledge become lower. Datasets which have been imported into the KGs, such as MusicBrainz into Freebase, have a huge impact on the number of triples and on the number of facts in the KG. Also the way of modeling data has a great impact on the number of triples. For instance, if n-ary relations are expressed in N-Triples format (as in case of Wikidata), many intermediate nodes need to be modeled, leading to many additional triples compared to plain statements. Last but not least, the number of supported languages influences the number of triples.
2. *Classes*: The number of classes is highly varying among the KGs, ranging from 736 (DBpedia) up to 300K (Wikidata) and 570K (YAGO). Despite its high number of classes, YAGO contains in relative terms the most classes which are actually used (i.e., classes with at least one instance). This can be traced back to the fact that heuristics are used for selecting appropriate Wikipedia categories as classes for YAGO. Wikidata, in contrast, contains many classes, but out of them only a small fraction

is actually used on instance level. Note, however, that this is not necessarily a burden.

3. *Domains*: Although all considered KGs are specified as crossdomain, domains are not equally distributed in the KGs. Also the domain coverage among the KGs differs considerably. Which domains are well represented heavily depends on which datasets have been integrated into the KGs. MusicBrainz facts had been imported into Freebase, leading to a strong knowledge representation (77%) in the domain of *media* in Freebase. In DBpedia and YAGO, the *domain* people is the largest, likely due to Wikipedia as data source.
4. *Relations and Predicates*: Many relations are rarely used in the KGs: Only 5% of the Freebase relations are used more than 500 times and about 70% are not used at all. In DBpedia, half of the relations of the DBpedia ontology are not used at all and only a quarter of the relations is used more than 500 times. For OpenCyc, 99.2% of the relations are not used. We assume that they are used only within Cyc, the commercial version of OpenCyc.
5. *Instances and Entities*: Freebase contains by far the highest number of entities. Wikidata exposes relatively many instances in comparison to the entities, as each statement is instantiated leading to around 74M instances which are not entities.
6. *Subjects and Objects*: YAGO provides the highest number of unique subjects among the KGs and also the highest ratio of the number of unique subjects to the number of unique objects. This is due to the fact that N-Quad representations need to be expressed via intermedium nodes and that YAGO is concentrated on classes which are linked by entities and other classes, but which do not provide outlinks. DBpedia exhibits more unique objects than unique subjects, since it contains many `owl:sameAs` statements to external entities.

5.2. Data Quality Analysis

We now present the results obtained by applying the DQ metrics introduced in the Sections 3.2 – 3.5 to the KGs DBpedia, Freebase, OpenCyc, Wikidata, and YAGO.

5.2.1. Accuracy

The fulfillment degrees of the KGs regarding the *Accuracy* metrics are shown in Table 3.

Table 3
Evaluation results for the KGs regarding the dimension *Accuracy*.

	DB	FB	OC	WD	YA
m_{synRDF}	1	1	1	1	1
m_{synLit}	0.99	1	1	1	0.62
$m_{semTriple}$	0.99	<1	1	0.99	0.99

Syntactic validity of RDF documents, m_{synRDF}

Evaluation method. For evaluating the *Syntactic validity of RDF documents*, we dereference the entity "Hamburg" as resource sample in each KG. In case of DBpedia, YAGO, Wikidata, and OpenCyc, there are RDF/XML serializations of the resource available, which can be validated by the official W3C RDF validator.[76] Freebase only provides a Turtle serialization. We evaluate the syntactic validity of this Turtle document by verifying if the document can be loaded into an RDF model of the Apache Jena Framework.[77]

Evaluation result. All considered KGs provide syntactically valid RDF documents. In case of YAGO and Wikidata, the RDF validator declares the used language codes as invalid, since the validator evaluates language codes in accordance with ISO-639. The criticized language codes are, however, contained in the newer standard ISO 639-3 and actually valid.

Syntactic validity of literals, m_{synLit}

Evaluation method. We evaluate the *Syntactic validity of literals* by means of the relations *date of birth*, *number of inhabitants*, and *International Standard Book Number (ISBN)*, as those relations cover different domains – namely, people, cities, and books – and as they can be found in all KGs. In general, domain knowledge is needed for selecting representative relations, so that a meaningful coverage is guaranteed.

Note that OpenCyc is not taken into account for this criterion: Although OpenCyc comprises around 1.1M literals in total, these literals are essentially labels and descriptions (given via `rdfs:label` and `rdfs:comment`), i.e., not aligned to specific data types. Hence, OpenCyc has no syntactic invalid literals and is assigned the metric value 1.

As long as a literal with data type is given, its syntax is verified with the help of the function `RDFDatatype .isValid(String)` of the Apache Jena framework.

[76] See `https://w3.org/RDF/Validator/`, requested on Mar 2, 2016.

[77] See `https://jena.apache.org/`, requested Mar 2, 2016.

Thereby, standard data types such as `xsd:date` can be validated easily, especially if different data types are provided.[78] If no data type is provided or if the literal value is of type `xsd:String`, the literal is evaluated by a regular expression, which is created manually (see below, depending on the considered relation). For each of the three relations we created a sample of 1M literal values per KG, as long as the respective KG contains so many literals.

Evaluation results. All KGs except YAGO performed very well regarding the *Syntactic validity of literals*.

Date of Birth For Wikidata, DBpedia, and Freebase, all verified literal values (1M per KG) were syntactically correct.[79] For YAGO, we detected around 519K syntactic errors (given 1M literal values) due to the usage of wildcards in the date values. For instance, the birth date of `yago:Socrates` is specified as "470-##-##", which does not correspond to the syntax of `xsd:date`. Obviously, the syntactic invalidity of literals is accepted by the YAGO publishers in order to keep the number of relations low.[80]

Number of inhabitants The data types of the literal values regarding the number of inhabitants were valid in all KGs. For DBpedia, YAGO, and Wikidata, we evaluated the syntactic validity of the number of inhabitants by checking if `xsd:nonNegativeInteger`, `xsd:decimal`, and `xsd:integer` were used as data types for the typed literals. In Freebase, no data type is specified. Therefore, we evaluated the values by means of a regular expression which allows only the decimals 0-9, periods, and commas.

ISBN The ISBN is an identifier for books and magazines. The identifier can occur in various formats: with or without preceding "ISBN," with or without delimiters, and with 10 or 13 digits. Gupta[81] provided a regular expression for validating ISBN in its different forms, which we used in our evaluation. All in all, most of the ISBN were assessed as syntactically correct. The lowest fulfillment degree was obtained for DBpedia. We found the following findings for the single KGs: In Freebase, around 699K ISBN numbers were available. Out of them, 38 were assessed as syntactically incorrect. Typical mistakes were too long numbers and wrong prefixes.[82] In case of Wikidata, 18 of around 11K ISBN numbers were syntactically invalid. However, some invalid numbers have meanwhile been corrected. This indicates that the Wikidata community does not only care about inserting new data, but also about curating given KG data. In case of YAGO, we could only find 400 triples with the relation `yago:hasISBN`. Seven of the literals on the object position were syntactically incorrect. For DBpedia, we evaluated around 24K literals. 7,419 of them were assessed as syntactically incorrect. In many cases, comments next to the ISBN numbers in the info-boxes of Wikipedia led to an inaccurate extraction of data, so that the comments are either extracted as additional facts about ISBN numbers[83] or together with the actual ISBN numbers as coherent strings.[84]

Semantic validity of triples, $m_{semTriple}$

Evaluation method. The semantic validity can be reliably measured by means of a reference data set which (i) contains at least to some degree the same facts as in the KG and (ii) which is regarded as some kind of authority. We decided to use the Integrated Authority File (Gemeinsame Normdatei, GND),[85] which is an authority file, especially concerning persons and corporate bodies, and which was created manually by German libraries. Due to the focus on persons (especially authors), we decided to evaluate a random sample of person entities w.r.t. the following relations: *birth place*, *death place*, *birth date*, and *death date*. For each of these relations, the corresponding relations in the KGs were determined. Then, a random sample of 100 person entities per KG was chosen. For each entity we retrieved the facts with the mentioned relations and assessed manually whether a GND entry exists and whether the values of the relations match with the values in the KG.

Evaluation result. We evaluated up to 400 facts per KG and observed only for a few facts some discrepancies. For instance, Wikidata states as death date of

[78] In DBpedia, for instance, data for the relation `dbo:birthDate` is stored both as `xsd:gYear` and `xsd:date`.

[79] Surprisingly, the Jena Framework assessed data values with a negative year (i.e., B.C.; e.g., "-600" for `xsd:gYear`) as invalid, despite the correct syntax.

[80] In order to model the dates to the extent they are known, further relations would be necessary, such as using `:wasBornOnYear` with range `xsd:gYear`, `:wasBornOnYearMonth` with range `xsd:gYearMonth`.

[81] See `http://howtodoinjava.com/regex/java-regex-validate-international-standard-book-number-isbns/`, requested on Mar 1, 2016.

[82] E.g., we found the 16 digit ISBN *9789780307986931* (cf. `freebase:m.0pkny27`) and the ISBN *2940045143431* with prefix *294* instead of *978* (cf. `freebase:m.0v3xf7b`).

[83] See `dbr:Prince_Caspian`.

[84] An example is "ISBN 0755111974 (hardcover edition)" for `dbr:My_Family_and_Other_Animals`.

[85] See `http://www.dnb.de/EN/Standardisierung/GND/gnd.html`, requested on Sep 8, 2016.

"Anton Erkelenz" (wdt:Q589196) April 24, whereas GND states April 25. For DBpedia and YAGO we encountered 3 and for Wikidata 4 errors. Hence, those KGs were evaluated with 0.99. Note that OpenCyc has no values for the chosen relations and thus evaluates to 1.

During evaluation we identified the following issues:

1. For finding the right entry in GND, more information besides the name of the person is needed. This information is sometimes not given, so that entity disambiguation is in those cases hard to perform.
2. Contrary to assumptions, often either no corresponding GND entry exists or not many facts of the GND entity are given. In other words, GND is incomplete w.r.t. to entities (cf. *Population completeness*) and relations (cf. *Column completeness*).
3. Values of different granularity need to be matched, such as an exact date of birth against the indication of a year only.

In conclusion, the evaluation of semantic validity is hard, even if a random sample set is evaluated manually. Meaningful differences among the KGs might be revealed only when a very large sample is evaluated, e.g., by using crowd-sourcing [2,3,48]. Another approach for assessing the semantic validity is presented by Kontokostas et al. [34] who propose a test-driven evaluation where test cases are created to evaluate triples semi-automatically: For instance, an interval specifies the valid height of a person and all triples which lie outside of this interval are evaluated manually. In this way, outliers can be easily found but possible wrong values within the interval are not detected.

Our findings appear to be consistent with the evaluation results of the YAGO developer team for YAGO2, where manually assessing 4,412 statements resulted in an accuracy of 98.1%.[86]

5.2.2. Trustworthiness

The fulfillment degrees of the KGs regarding the *Trustworthiness* criteria are shown in Table 4.

Table 4
Evaluation results for the KGs regarding the dimension *Trustworthiness*.

	DB	FB	OC	WD	YA
m_{graph}	0.5	0.5	1	0.75	0.25
m_{fact}	0.5	1	0	1	1
m_{NoVal}	0	1	0	1	0

Trustworthiness on KG level, m_{graph}

Evaluation method. Regarding the trustworthiness of a KG in general, we differentiate between the method of how new data is inserted into the KG and the method of how existing data is curated.

Evaluation results. The KGs differ considerably w.r.t. this metric. OpenCyc obtains the highest score here, followed by Wikidata. In the following, we provide findings for the single KGs, which are listed by decreasing fulfillment score:

Cyc is edited (expanded and modified) exclusively by a dedicated expert group. The free version, OpenCyc, is derived from Cyc and only a locally hosted version can be modified by the data consumer.

Wikidata is also curated and expanded manually, but by volunteers of the Wikidata community. Wikidata allows importing data from external sources such as Freebase.[87] However, new data is not just inserted, but is approved by the community.

Freebase was also curated by a community of volunteers. In contrast to Wikidata, the proportion of data imported automatically is considerably higher and new data imports were not dependent on community approvals.

DBpedia and YAGO The knowledge of both KGs is extracted from Wikipedia, but DBpedia differs from YAGO w.r.t. the community involvement: Any user can engage (i) in mapping the Wikipedia infobox templates to the DBpedia ontology in the DBpedia mappings wiki[88] and (ii) in the development of the DBpedia extraction framework.

Trustworthiness on statement level

We determine the *Trustworthiness on statement level* by evaluating whether provenance information for statements is used in the KGs. The picture is mixed:

DBpedia uses the relation prov:wasDerived From to store the sources of the entities and their state-

[86] With a weighted averaging of 95%, see http://www.mpi-inf.mpg.de/de/departments/databases-and-information-systems/research/yago-naga/yago/statistics/, requested on Mar 3, 2016.

[87] Note that imports from Freebase require the approval of the community (see https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool). Besides that, there are bots which import automatically (see https://www.wikidata.org/wiki/Wikidata:Bots/de).

[88] See http://mappings.dbpedia.org/, requested on Mar 3, 2016.

ments. However, as the source is always the corresponding Wikipedia article,[89] this provenance information is trivial and the fulfillment degree is, hence, of rather formal nature.

YAGO uses its own vocabulary to indicate the source of information. Interestingly, YAGO stores per statement both the source (via `yago:extractionSource`; e.g., the Wikipedia article) and the used extraction technique (via `yago:extractionTechnique`; e.g., "Infobox Extractor" or "CategoryMapper"). The number of statements about sources is 161M, and, hence, many times over the number of instances in the KG. The reason for that is that in YAGO the source is stored for each fact.

In *Wikidata* several relations can be used for referring to sources, such as "imported from" (`wdt:P143`), "stated in" (`wdt:P248`), and "reference URL" (`wdt:P854`).[90] Note that "imported from" relations are used for automatic imports but that statements with such a reference are not accepted ("data is not sourced").[91] To source data, the other relations, "stated in" and "reference URL", can be used. The number of all stored references in Wikidata[92] is around 971K. Based on the number of all statements,[93] 74M, this corresponds to a coverage of around 1.3%. Note, however, that not every statement in Wikidata requires a reference according to the Wikidata guidelines. In order to be able to state how many references are actually missing, a manual evaluation would be necessary. However, such an evaluation would be presumably highly subjective.

Freebase uses proprietary vocabulary for representing provenance: via *n-ary relations*, which are in Freebase called *Compound Value Types (CVT)*, data from higher arity can be expressed [44].[94]

OpenCyc differs from the other KGs in that it uses neither an external vocabulary nor a proprietary vocabulary for storing provenance information.

[89] E.g., `http://en.wikipedia.org/wiki/Hamburg` for `dbr:Hamburg`.

[90] All relations are instances of "Wikidata property to indicate a source" (`wdt:Q18608359`).

[91] See `https://www.wikidata.org/wiki/Property:P143`, requested Mar 3, 2016.

[92] This is the number of instances of `wdo:Reference`.

[93] This is the number of instances of `wdo:Statement`.

[94] E.g., for a statement with the relation `freebase:location.statistical_region.population`, the source can be stored via `freebase:measurement_unit.dated_integer.source`.

Table 5
Evaluation results for the KGs regarding the dimension *Consistency*.

	DB	FB	OC	WD	YA
$m_{checkRestr}$	0	1	0	1	0
$m_{conClass}$	0.88	1	<1	1	0.33
$m_{conRelat}$	0.99	0.45	1	0.50	0.99

Indicating unknown and empty values, m_{NoVal}

This criterion highlights the subtle data model of Wikidata and Freebase in comparison to the data models of the other KGs: Wikidata allows for storing unknown values and empty values (e.g., that "Elizabeth I of England" (`wdt:Q7207`) had no children). However, in the Wikidata RDF export such statements are only indirectly available, since they are represented via blank nodes and via the relation `owl:someValuesFrom`.

YAGO supports the representation of unknown values and empty values by providing explicit relations for such cases.[95] Inexact dates are modeled by means of wildcards (e.g., "1940-##-##", if only the year is known). Note, however, the invalidity of such strings as date literals (see Section 5.2.1). Unknown dates are not supported by YAGO.

5.2.3. Consistency

The fulfillment degrees of the KGs regarding the *Consistency* criteria are shown in Table 5.

Check of schema restrictions during insertion of new statements, $m_{checkRestr}$

The values of the metric $m_{checkRestr}$, indicating restrictions during the insertion of new statements, are varying among the KGs. The web interfaces of Freebase and Wikidata verify during the insertion of new statements by the user whether the input is compatible with the respective data type. For instance, data of the relation "date of birth" (`wdt:P569`) is expected to be in a syntactically valid form. DBpedia, OpenCyc and YAGO have no checks for schema restriction during the insertion of new statements.

Consistency of statements w.r.t. class constraints, $m_{conClass}$

Evaluation method. For evaluating the consistency of class constraints we considered the relation `owl:disjointWith`, since this is the only relation which is used by more than half of the consid-

[95] E.g., `freebase:freebase.valuenotation.has_no_value`.

ered KGs. We only focused on direct instantiations here: if there is, for instance, the triple (dbo:Plant, owl:disjointWith, dbo:Animal), then there must not be a resource which is instantiated both as dbo:Plant and dbo:Animal.

Evaluation results. We obtained mixed results here. Only Freebase, OpenCyc, and Wikidata perform very well.[96]

Freebase and Wikidata do not specify any constraints with owl:disjointWith. Hence, those two KGs have no inconsistencies w.r.t. class restrictions and we can assign the metric value 1 to them. In case of OpenCyc, 5 out of the 27,112 class restrictions are inconsistent. DBpedia contains 24 class constraints. Three out of them are inconsistent. For instance, over 1,200 instances exist which are both a dbo:Agent and a dbo:Place. YAGO contains 42 constraints, dedicated mainly for WordNet classes, which are mostly inconsistent.

Consistency of statements w.r.t. relation constraints, $m_{conRelat}$

Evaluation method Here we considered the relations rdfs:range and owl:FunctionalProperty, as those are used in more than every second considered KG. rdfs:range specifies the expected type of an instance on the object position of a triple, while owl:FunctionalProperty indicates that a relation should only be used at most once per resource. We only took datatype properties into account for this evaluation, since consistencies regarding object properties would require to distinguish Open World assumption and Closed World assumption.

Evaluation results. In the following, we consider the fulfillment degree for the relation constraints rdfs:range and owl:FunctionalProperty separately. In Table 5, we show the average of the fulfillment scores of each KG regarding rdfs:range and owl:FunctionalProperty. Note that the numbers of evaluated relation constraints varied from KG to KG, depending on how many relation constraints were available per KG.

Range. *Wikidata* does not use any rdfs:range restrictions. Within the Wikidata data model, there is wdo:propertyType, but this indicates not the exact allowed data type of a relation (e.g., wdo:propertyTypeTime can represent a year or an exact date). On the talk pages of Wikidata relations users can indicate the allowed values of relations via "One of" statements.[97] Since "One of" statements are only listed on the property talk pages and since not only entity types but also concrete instances are used as "One of" values, we do not consider those statements here.

Table 6
Evaluation results for the KGs regarding the dimension *Relevancy*.

	DB	FB	OC	WD	YA
$m_{Ranking}$	0	1	0	1	0

DBpedia obtains the highest measured fulfillment score w.r.t. consistency of rdfs:range statements. An example for a range inconsistency is that the relation dbo:birthDate requires a data type xsd:date; in about 20% of those relations, the data type xsd:gYear is used, though.

YAGO, *Freebase*, and *OpenCyc* contain range inconsistencies primarily since they specify designated data types via range relations which are not consistently used on the instance level. For instance, YAGO specifies proprietary data types such as yago:yagoURL and yago:yagoISBN. On the instance level, however, either no data type is used or the unspecific data type xsd:string.

FunctionalProperty. The restriction indicated by owl:FunctionalProperty is used by all KGs except Wikidata. On the talk pages about the relations in Wikidata, users can specify the cardinality restriction via setting the relation to "single"; however, this is not part of the Wikidata data model. The other KGs mostly comply with the usage restrictions of owl:FunctionalProperty. Noteworthy is that in Freebase 99.9% of the inconsistencies obtained here are caused by the usages of the relations freebase:type.object.name and freebase:common.notable_for.display_name.

5.2.4. Relevancy

The fulfillment degrees of the KGs regarding the *Relevancy* criteria are shown in Table 6.

Creating a ranking of statements, $m_{Ranking}$

Only Wikidata supports the modeling of a ranking of statements: Each statement is ranked with “pre-

[96] Note that the sample size varies among the KGs (depending on how many owl:disjointWith statements are available per KG). Therefore, inconsistencies measured on a small set of owl:disjointWith facts become more visible.

[97] See https://www.wikidata.org/wiki/Category:Properties_with_one-of_constraints for an overview; requested on Jan 29, 2017.

Table 7
Evaluation results for the KGs regarding the dimension *Completeness*.

	DB	FB	OC	WD	YA
$m_{cSchema}$	0.91	0.76	0.92	1	0.95
$m_{cColumn}$	0.40	0.43	0	0.29	0.33
m_{cPop}	0.93	0.94	0.48	0.99	0.89
m_{cPop} (short)	1	1	0.82	1	0.90
m_{cPop} (long)	0.86	0.88	0.14	0.98	0.88

ferred rank" (`wdo:PreferredRank`), "normal rank" (`wdo:NormalRank`), or "deprecated rank" (`wdo:DeprecatedRank`). The "preferred rank" corresponds to the up-to-date value or the consensus of the Wikidata community w.r.t. this relation. Freebase does not provide any ranking of statements, entities, or relations. However, the meanwhile shutdown Freebase Search API provided a ranking for resources.[98]

5.2.5. Completeness

The fulfillment degrees of the KGs regarding the *Completeness* criteria are shown in Table 7.

Schema completeness, $m_{cSchema}$

Evaluation method. Since a gold standard for evaluating the *Schema completeness* of the considered KGs has not been published, we built one on our own. This gold standard is available online.[99] It is based on the data set used in Section 5.1.3, where we needed assignments of classes to domains, and comprises of 41 classes as well as 22 relations. It is oriented towards the domains *people*, *media*, *organizations*, *geography*, and *biology*. The classes in the gold standard were aligned to corresponding WordNet synsets (using WordNet version 3.1) and were grouped into main classes.

Evaluation results. Generally, Wikidata performs optimal; also DBpedia, OpenCyc, and YAGO exhibit results which can be judged as acceptable for most use cases. Freebase shows considerable room for improvement concerning the coverage of typical cross-domain classes and relations. The results in more detail are as follows:

DBpedia. DBpedia shows a good score regarding *Schema completeness* and its schema is mainly limited due to the characteristics of how information is stored and extracted from Wikipedia.

1. Classes: The DBpedia ontology was created manually and covers all domains well. However, it is incomplete in the details and therefore appears unbalanced. For instance, within the domain of plants the DBpedia ontology does not use the class "tree" but the class "ginko," which is a subclass of trees. We can mention as reason for such gaps in the modeling the fact that the ontology is created by means of the most frequently used infobox templates in Wikipedia.

2. Relations: Relations are considerably well covered in the DBpedia ontology. Some missing relations or modeling failures are due to the Wikipedia infobox characteristics. For example, to represent the gender of a person the existing relation `foaf:gender` seems to fit. However, it is only modeled in the ontology as belonging to the class `dbo:language` and not used on instance level. Note that the gender of a person is often not explicitly mentioned in the Wikipedia infoboxes but implicitly mentioned in the category names (for instance, "American male singers"). While DBpedia does not exploit this knowledge, YAGO does use it and provides facts with the relation `yago:hasGender`.

Freebase. Freebase shows a very ambivalent schema completeness. On the one hand, Freebase targets rather the representation of facts on instance level than the representation of classes and their hierarchy. On the other hand, Freebase provides a vast amount of relations, leading to a very good coverage of the requested relations.

1. Classes: Freebase lacks a class hierarchy and subclasses of classes are often in different domains (for instance, the classes `freebase:music.artist` and sportsmen `freebase:sports.pro_athlete` are logically a subclass of the class people `freebase:person.people` but not explicitly stated as such), which makes it difficult to find suitable sub- and superclasses. Noteworthy, the biology domain contains no classes. This is due to the fact that classes are represented as entities, such as *tree*[100] and *ginko*.[101] The ginko tree is not classified as tree, but by the generic class `freebase:biology.oganism_classification`.

2. Relations: Freebase exhibits all relations requested by our gold standard. This is not surprising, given the vast amount of available relations in Freebase (see Section 5.1.4 and Table 2).

[98] See `https://developers.google.com/freebase/v1/search-cookbook#scoring-and-ranking`, requested on Mar 4, 2016.

[99] See `http://km.aifb.kit.edu/sites/knowledge-graph-comparison/`, requested on Jan 29, 2017.

[100] Freebase ID `freebase:m.07j7r`.

[101] Freebase ID `freebase:m.0htd3`.

OpenCyc. In total, OpenCyc exposes a quite high *Schema completeness* scoring. This is due to the fact that OpenCyc has been created manually and has its focus on generic and common-sense knowledge.

1. Classes: The ontology of OpenCyc covers both generic and specific classes such as `cych:SocialGroup` and `cych:LandTopographicalFeature`. We can state that OpenCyc is complete with respect to the considered classes.

2. Relations: OpenCyc lacks some relations of the gold standard such as the number of pages or the ISBN of books.

Wikidata. According to our evaluation, Wikidata is complete both with respect to classes and relations.

1. Classes: Besides frequently used generic classes such as "human" (`wdt:Q5`) also very specific classes exist such as "landform" (`wdt:Q271669`) in the sense of a geomorphologial unit with over 3K instances.

2. Relations: In particular remarkable is that Wikidata covers all relations of the gold standard, even though it has extremely less relations than Freebase. Thus, the Wikidata methodology to let users propose new relations, to discuss about their outreach, and finally to approve or disapprove the relations, seems to be appropriate.

YAGO. Due to its concentration on modeling classes, YAGO shows the best overall *Schema completeness* fulfillment score among the KGs.

1. Classes: To create the set of classes in YAGO, the Wikipedia categories are extracted and connected to WordNet synsets. Since also our gold standard is already aligned to WordNet synsets, we can measure a full completeness score for YAGO classes.

2. Relations: The YAGO schema does not contain many unique but rather abstract relations, which can be understood in different senses. The abstract relation names make it often difficult to infer the meaning. The relation `yago:wasCreatedOnDate`, for instance, can be used reasonably for both the foundation year of a company and for the publication date of a movie. DBpedia, in contrast, provides the relation `dbp:foundationYear`. Often the meaning of YAGO relations is only fully understood after considering the associated classes, using domain and range of the relations. Expanding the YAGO schema by further, more fine-grained relations appears reasonable.

Column completeness, $m_{cColumn}$

Evaluation method. For evaluating KGs w.r.t. *Column completeness*, for each KG 25 class-relation-combinations[102] were created based on our gold standard created for measuring the *Schema completeness*. It was ensured that only those relations were selected for a given class for which a value typically exists for that class. For instance, we did not include the death date as potential relation for living people.

Table 8
Metric values of m_{cCol} for single class-relation-pairs.

Relation	DB	FB	OC	ED	YA
Person–birthdate	0.48	0.48	0	0.70	0.77
Person–sex	–	0.57	0	0.94	0.64
Book–author	0.91	0.93	0	0.82	0.28
Book–ISBN	0.73	0.63	–	0.18	0.01

Evaluation results. In general, no KG yields a metric score of over 0.43. As visible in Table 8, KGs often have some specific class-relation-pairs which are well represented on instance level, while the rest of the pairs are poorly represented. The well-represented pairs presumably originate either from column-complete data sets which were imported (cf. MusicBrainz in case of Freebase), or from user edits focusing primarily on facts about entities of popular classes such as people. We notice the following observations with respect to the single KGs:

DBpedia. DBpedia fails regarding the relation *sex* for instances of class `Person`, since it does not contain such a relation in its ontology. If we considered the non-mapping-based property `dbp:gender` instead (not defined in the ontology), we would gain a coverage of only 0.25% (about 5K people). We can note, hence, that the extraction of data out of the Wikipedia categories would be a further fruitful data source for DBpedia.

Freebase. Freebase surprisingly shows a very high coverage (92.7%) of the authors of books, given the basic population of 1.7M books. Note, however, that there are not only books modeled under `freebase:book.book` but also entities of other types, such as a description of the Lord of Rings (see `freebase:m.07bz5`). Also the coverage of ISBN for books is quite high (63.4%).

OpenCyc. OpenCyc breaks ranks, as mostly no values for the considered relations are stored in this KG. It

[102] The selection of class-relation-pairs was depending on the fact which class-relation-pairs were available per KG. Hence, the choice is varying from KG to KG. Also, note that less class-relation-pairs were used if no 25 pairs were available in the respective KG.

contains mainly taxonomic knowledge and only thinly spread instance facts.

Wikidata. Wikidata achieves a high coverage of birth dates (70.3%) and of gender (94.1%), despite the high number of 3M people.[103]

YAGO. YAGO obtains a coverage of 63.5% for gender relations, as it, in contrast to DBpedia, extracts this implicit information from Wikipedia.

Population completeness, m_{cPop}

Evaluation method. In order to evaluate the *Population completeness*, we need a gold standard consisting of a basic entity population for each considered KG. This gold standard, which is available online,[104] was created on the basis of our gold standard used for evaluating the *Schema completeness* and the *Column completeness*. For its creation, we selected five classes from each of the five domains and determined two well-known entities (called "short head") and two rather unknown entities (called "long tail") for each of those classes. The exact entity selection criteria are as follows.

1. The well-known entities were chosen without temporal and location-based restrictions. To take the most popular entities per domain, we used quantitative statements. For instance, to select well-known athletes, we ranked athletes by the number of won olympic medals; to select the most popular mountains, we ranked the mountains by their heights.
2. To select the rather unknown entities, we considered entities associated to both Germany and a specific year. For instance, regarding the athletes, we selected German athletes active in the year 2010, such as Maria Höfl-Riesch. The selection of rather unknown entities in the domain of biology is based on the IUCN Red List of Threatened Species[105].[106]

Selecting four entities per class and five classes per domain resulted in 100 entities to be used for evaluating the *Population completeness*.

[103] These 3M instances form about 18.5% of all instances in Wikidata. See `https://www.wikidata.org/wiki/Wikidata:Statistics`, requested on Nov 7, 2016.

[104] See `http://km.aifb.kit.edu/sites/knowledge-graph-comparison/`, requested on Jan 29, 2017.

[105] See `http://www.iucnredlist.org`, requested on Apr 2, 2016.

[106] Note that selecting entities by their importance or popularity is hard in general and that also other popularity measures such as the PageRank scores may be taken into account.

Evaluation results. All KGs except OpenCyc show good evaluation results. Since also Wikidata exhibits good evaluation results, the population degree apparently does not depend on the age or the insertion method of the KG. Fig. 10 additionally depicts the population completeness for the single domains for each KG. In the following, we firstly present our findings for well-known entities, before we secondly go into the details of rather unknown entities.

Well-known entities: Here, all considered KGs achieve good results. DBpedia, Freebase, and Wikidata are complete w.r.t. the well-known entities in our gold standard. YAGO lacks some well-known entities, although some of them are represented in Wikipedia. One reason for this fact is that those Wikipedia entities do not get imported into YAGO for which a WordNet class exists. For instance, there is no “Great White Shark” entity, only the WordNet class `yago:wordnet_great_white_shark_101484850`.

Not-well-known entities: First of all, not very surprising is the fact that all KGs show a higher degree of completeness regarding well-known entities than regarding rather unknown entities, as the KGs are oriented towards general knowledge and not domain-specific knowledge. Secondly, two things are in particular peculiar concerning long-tail entities in the KGs: While most of the KGs obtain a score of about 0.88, Wikidata deflects upwards and OpenCyc deflects strongly downwards.

Wikidata exhibits a very high *Population completeness* degree for long tail entities. This is a result from the central storage of interwiki links between different Wikimedia projects (especially between the different Wikipedia language versions) in Wikidata: A Wikidata entry is added to Wikidata as soon as a new entity is added in one of the many Wikipedia language versions. Note, however, that in this way English-language labels for the entities are often missing. We measure that only about 54.6% (10.2M) of all Wikidata resources have an English label.

OpenCyc exhibits a poor population degree score of 0.14 for long-tail entities. OpenCyc’s sister KGs Cyc and ResearchCyc are apparently considerably better covered with entities [36], leading to higher *Population completeness* scores.

5.2.6. Timeliness

The evaluation results concerning the dimension *Timeliness* are presented in Table 9.

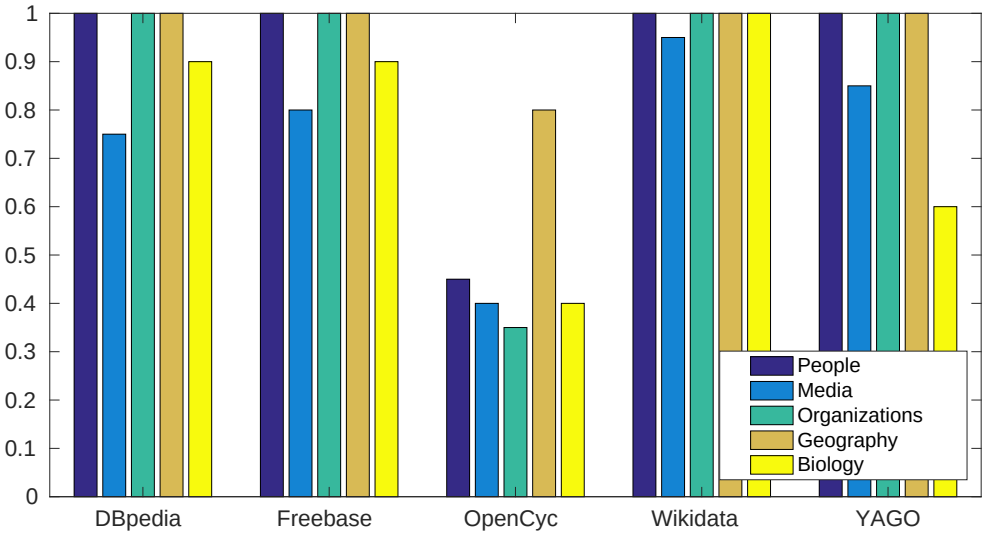

Fig. 10. Population completeness regarding the different domains per KG.

Table 9
Evaluation results for the KGs regarding the dimension *Timeliness*.

	DB	FB	OC	WD	YA
m_{Freq}	0.5	0	0.25	1	0.25
$m_{Validity}$	0	1	0	1	1
m_{Change}	0	1	0	0	0

Timeliness frequency of the KG, m_{Freq}

Evaluation results. The KGs are very diverse regarding the frequency in which the KGs are updated, ranging from a score of 0 for Freebase (not updated any more) to 1 for Wikidata (updates immediately visible and retrievable). Note that the *Timeliness frequency of the KG* can be a crucial point and a criterion for exclusion in the process of choosing the right KG for a given setting [17]. In the following, we outline some characteristics of the KGs with respect to their up-to-dateness:

DBpedia is created about once to twice a year and is not modified in the meantime. From September 2013 until November 2016, six DBpedia versions have been published.[107] Besides the static DBpedia, DBpedia live[108] has been continuously updated by tracking changes in Wikipedia in real-time. However, it does not provide the full range of relations as DBpedia.

Freebase had been updated continuously until its close-down and is not updated anymore.

OpenCyc has been updated less than once per year. The last OpenCyc version dates from May 2012.[109] To the best of our knowledge, Cyc and OpenCyc, respectively, are developed further, but no exact date of the next version is known.

Wikidata provides the highest fulfillment degree for this criterion. Modifications in Wikidata are via browser and via HTTP URI dereferencing immediately visible. Hence, Wikidata falls in the category of continuous updates. Besides that, an RDF export is provided on a roughly monthly basis (either via the RDF export webpage[110] or via own processing using the Wikidata toolkit[111]).

YAGO has been updated less than once per year. YAGO3 was published in 2015, YAGO2 in 2011, and the interim version YAGO2s in 2013. A date of the next release has not been published.

Specification of the validity period of statements, $m_{Validity}$

Evaluation results. Although representing the validity period of statements is obviously reasonable for many relations (for instance, the president's term of

[107] These versions are DBpedia 3.8, DBpedia 3.9, DBpedia 2014, DBpedia 2015-04, DBpedia 2015-10, and DBpedia 2016-04. Always the latest DBpedia version is published online for dereferencing.

[108] See `http://live.dbpedia.org/`, requested on Mar 4, 2016.

[109] See `http://sw.opencyc.org/`, requested on Nov 8, 2016.

[110] See `http://tools.wmflabs.org/wikidata-exports/rdf/exports/`, requested on Nov 23, 2016.

[111] See `https://github.com/Wikidata/Wikidata-Toolkit`, requested on Nov 8, 2016.

Table 10
Evaluation results for the KGs regarding the dimension *Ease of understanding*.

	DB	FB	OC	WD	YA
m_{Descr}	0.70	0.97	1	<1	1
m_{Lang}	1	1	0	1	1
m_{uSer}	1	1	0	1	1
m_{uURI}	1	0.5	1	0	1

office), specifying the validity period of statements is in several KGs either not possible at all or only rudimentary performed.

DBpedia and OpenCyc do not realize any specification possibility. In YAGO, Freebase, and Wikidata the temporal validity period of statements can be specified. In YAGO, this modeling possibility is made available via the relations `yago:occursSince`, `yago:occursUntil`, and `yago:occursOnDate`. Wikidata provides the relations "start time" (`wdt:P580`) and "end time" (`wdt:P582`). In Freebase, Compound Value Types (CVTs) are used to represent relations with higher arity [44]. As part of this representation, validity periods of statements can be specified. An example is "Vancouver's population in 1997."

Specification of the modification date of statements, m_{Change}

Evaluation results. The modification date of statements can only be specified in Freebase but not in the other KGs. Together with the criteria on *Timeliness*, this reflects that the considered KGs are mostly not sufficiently equipped with possibilities for modeling temporal aspects within and about the KG.

In Freebase the date of the last review of a fact can be represented via the relation `freebase:freebase.valuenotation.is_reviewed`. In the DBpedia ontology the relation `dcterms:modified` is used to state the date of the last revision of the DBpedia ontology. When dereferencing a resource in Wikidata, the latest modification date of the resource is returned via `schema:dateModified`. This, however, does not hold for statements. Thus Wikidata is evaluated with 0, too.

5.2.7. Ease of Understanding

Description of resources, m_{Descr}

Evaluation method. We measured the extent to which entities are described. Regarding the labels, we considered `rdfs:label` for all KGs. Regarding the descriptions, the corresponding relations differ from KG to KG: DBpedia, for instance, uses `rdfs:comment` and `dcelements:description`, while Freebase provides `freebase:common.topic.description`.[112]

Evaluation result. For all KGs the rule applies that in case there is no label available, usually there is also no description available. The current metric could therefore (without significant restrictions) be applied to `rdfs:label` occurrences only.

YAGO, Wikidata, and OpenCyc contain a label for almost every entity. In Wikidata, the entities without any label are of experimental nature and are most likely not used.[113]

Surprisingly, DBpedia shows a relatively low coverage w.r.t. labels and descriptions (only 70.4%). Our manual investigations suggest that relations with higher arity are modeled by means of intermediate nodes which have no labels.[114]

Labels in multiple languages, m_{Lang}

Evaluation method. Here we measure whether the KGs contain labels (`rdfs:label`) in other languages than English. This is done by means of the language annotations of literals such as "@de" for literals in German.

Evaluation results. *DBpedia* provides labels in 13 languages. Further languages are provided in the localized DBpedia versions. *YAGO* integrates statements of the different language versions of Wikipedia into one KG. Therefore, it provides labels in 326 different languages. *Freebase and Wikidata* also provide a lot of languages (244 and 395 languages, respectively). Contrary to the other KGs, *OpenCyc* only provides labels in English.

Coverage of languages. We also measured the coverage of selected languages in the KGs, i.e., the extent to which entities have an `rdfs:label` with a specific language annotation.[115] Our evaluation shows that DBpedia, YAGO, and Freebase achieve a high coverage with more than 90% regarding the English language. In contrast to those KGs, Wikidata shows a relative low

[112] Human-readable resource descriptions may also be represented by other relations [15]. However, we focused on those relations which are commonly used in the considered KGs.

[113] For instance, `wdt:Q5127809` represents a game fo the Nintendo Entertainment System, but there is no further information for an identification of the entity available.

[114] E.g., `dbr:Nayim` links via `dbo:CareerStation` to 10 entities of his carrier stations.

[115] Note that literals such as `rdfs:label` do not necessarily have language annotations. In those cases, we assume that no language information is available.

coverage regarding the English language of only 54.6%, but a coverage of over 30% for further languages such as German and French. Wikidata is, hence, not only the most diverse KG in terms of languages, but has also the highest coverage regarding non-English languages.

Understandable RDF serialization, m_{uSer}

The provisioning of understandable RDF serializations in the context of URI dereferencing leads to a better understandability for human data consumers. DBpedia, YAGO, and Wikidata provide N-Triples and N3/Turtle serializations. Freebase, in contrast, only provides a Turtle serialization. OpenCyc only uses RDF/XML, which is regarded as not easily understandable by humans.

Self-describing URIs, m_{uURI}

We can observe two different paradigms of URI usage: On the one hand, DBpedia, OpenCyc, and YAGO rely on descriptive URIs and therefore achieve the full fulfillment degree. In DBpedia and YAGO, the URIs of the entities are determined by the corresponding English Wikipedia article. The mapping to the English Wikipedia is thus trivial. In case of OpenCyc, two RDF exports are provided: one using opaque and one using self-describing URIs. The self-describing URIs are thereby derived from the `rdfs:label` values of the resources.

On the other hand, *Wikidata* and *Freebase* (the latter in part) rely on opaque URIs: Wikidata uses Q-IDs for resources ("items" in Wikidata terminology) and P-IDs for relations. Freebase uses self-describing URIs only partially, namely, opaque M-IDs for entities and self-describing URIs for classes and relations.[116]

5.2.8. Interoperability

The evaluation results of the dimension *Interoperability* are presented in Table 11.

Table 11
Evaluation results for the KGs regarding the dimension *Interoperability*.

	DB	FB	OC	WD	YA
m_{Reif}	0.5	0.5	0.5	0	0.5
$m_{iSerial}$	1	0	0.5	1	1
m_{extVoc}	0.61	0.11	0.41	0.68	0.13
$m_{propVoc}$	0.15	0	0.51	>0	0

Avoiding blank nodes and RDF reification, m_{Reif}

Reification allows to represent further information about single statements. In conclusion, we can state that DBpedia, Freebase, OpenCyc, and YAGO use some form of reification. However, none of the considered KGs uses the RDF standard for reification. Wikidata makes extensive use of reification: every relation is stored in the form of an *n-ary relation*. In case of DBpedia and Freebase, in contrast, facts are predominantly stored as N-Tripels and only relations of higher arity are stored via *n-ary relations*.[117] YAGO stores facts as *N-Quads* in order to be able to store meta information of facts like provenance information. When the quads are loaded in a triple store, the IDs referring to the single statements are ignored and quads are converted into triples. In this way, most of the statements are still usable without the necessity to deal with reification.

Blank nodes are non-dereferencable, anonymous resources. They are used by the Wikidata and OpenCyc data model.

Provisioning of several serialization formats, $m_{iSerial}$

DBpedia, YAGO, and Wikidata fulfill the criterion of *Provisioning several RDF serialization formats* to the full extent, as they provide data in RDF/XML and several other serialization formats during the URI dereferencing. In addition, DBpedia and YAGO provide further RDF serialization formats (e.g., JSON-LD, Microdata, and CSV) via their SPARQL endpoints. Freebase is the only KG providing RDF only in Turtle format.

Using external vocabulary, m_{extVoc}

Evaluation method. This criterion indicates the extent to which external vocabulary is used. For that, for each KG we divide the occurrence number of triples with external relations by the number of all relations in this KG.

Evaluation results. *DBpedia* uses 37 unique external relations from 8 different vocabularies, while the other KGs mainly restrict themselves to the external vocabularies RDF, RDFS, and OWL.

Wikidata reveals a high external vocabulary ratio, too. We can mention two obvious reasons for that fact: 1. Information in Wikidata is provided in a huge variety of languages, leading to 85M `rdfs:label` and 140M `schema:description` literals. 2. Wikidata makes extensive use of reification. Out of the 140M triples used for instantiations via `rdf:type`, about 74M (i.e.,

[116] E.g., `freebase:music.album` for the class "music albums" and `freebase:people.person.date_of_birth` for the relation "day of birth".

[117] See Section 5.1.1 for more details w.r.t. the influence of reification on the number of triples.

about the half) are taken for instantiations of statements, i.e., for reification.

Interoperability of proprietary vocabulary, $m_{propVoc}$
Evaluation method. This criterion determines the extent to which URIs of proprietary vocabulary are linked to external vocabulary via equivalence relations. For each KG, we measure which classes and relations are linked via `owl:sameAs`,[118] `owl:equivalentClass` (in Wikidata: `wdt:P1709`), and `owl:equivalentProperty` (in Wikidata `wdt:P1628`) to external vocabulary. Note that other relations such as `rdf:subPropertyOf` could be taken into account; however, in this work we only consider equivalency relations.

Evaluation results. In general, we obtained low fulfillment scores regarding this criterion. OpenCyc shows the highest value. We achieved the following single findings:

Regarding its classes, *DBpedia* reaches a relative high interlinking degree of about 48.4%. Classes are thereby linked to FOAF, Wikidata, schema.org and DUL.[119] Regarding its relations, DBpedia links to Wikidata and schema.org.[120] Only 6.3% of the DBpedia relations are linked to external vocabulary.

Freebase only provides `owl:sameAs` links in the form of a separate RDF file, but these links are only on instance level. Thus, the KG is evaluated with 0.

In *OpenCyc*, about half of all classes exhibit at least one external linking via `owl:sameAs`. Internal links to resources of `sw.cyc.com`, the commercial version of OpenCyc, were ignored in our evaluation. The considered classes are mainly linked to FOAF, UMBEL, DBpedia, and linkedmdb.org, the relations mainly to FOAF, DBpedia, Dublin Core Terms, and linkedmdb.org. The relative high linking degree of OpenCyc can be attributed to dedicated approaches of linking OpenCyc to other KGs (see, e.g., Medelyan et al. [38]).

Regarding the classes, *Wikidata* provides links mainly to DBpedia. Considering all Wikidata classes, only 0.1% of all Wikidata classes are linked to equivalent external classes. This may be due to the high number of classes in Wikidata in general. Regarding the relations, Wikidata provides links in particular to FOAF and schema.org and achieves here a linking coverage of 2.1%. Although this is low, frequently used relations are linked.[121]

YAGO contains around 553K `owl:equivalentClass` links to classes within the DBpedia namespace `dby:`. However, as YAGO classes (and their hierarchy) were imported also into DBpedia (using the namespace `http://dbpedia.org/class/yago/`), we do not count those `owl:equivalentClass` links in YAGO as external links for YAGO.

Table 12
Evaluation results for the KGs regarding the dimension *Accessibility*.

	DB	FB	OC	WD	YA
m_{Deref}	1	1	0.44	0.41	1
m_{Avai}	<1	0.73	<1	<1	1
m_{SPARQL}	1	1	0	1	0
m_{Export}	1	1	1	1	1
m_{Negot}	0.5	1	0	1	0
$m_{HTMLRDF}$	1	1	1	1	0
m_{Meta}	1	0	0	0	1

5.2.9. Accessibility

The evaluation results of the dimension *Accessibility* are presented in Table 12.

Dereferencing possibility of resources, m_{Deref}
Evaluation method. We measured the dereferencing possibilities of resources by trying to dereference URIs containing the fully-qualified domain name of the KG. For that, we randomly selected 15K URIs in the subject, predicate, and object position of triples in each KG. We submitted HTTP requests with the HTTP accept header field set to `application/rdf+xml` in order to perform content negotiation.

Evaluation results. In case of *DBpedia*, *OpenCyc*, and *YAGO*, all URIs were dereferenced successfully and returned appropriate RDF data, so that they fulfilled this criterion completely. For DBpedia, 45K URIs were analyzed, for OpenCyc only around 30K due to the small number of unique predicates. We observed almost

[118] OpenCyc uses `owl:sameAs` both on schema and instance level. This is appropriate as the OWL primer states "The built-in OWL property `owl:sameAs` links an individual to an individual" as well as "The `owl:sameAs` statements are often used in defining mappings between ontologies", see `https://www.w3.org/TR/2004/REC-owl-ref-20040210/#sameAs-def` (requested on Feb 4, 2017).

[119] See `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl`, requested on Jan 11, 2017.

[120] E.g., `dbo:birthDate` is linked to `wdt:P569` and `schema:birthDate`.

[121] Frequently used relations with stated equivalence to external relations are, e.g., `wdt:P31`, linked to `rdf:type`, and `wdt:P279`, linked to `rdfs:subClassOf`.

the same picture for YAGO, namely no notable errors during dereferencing.

For *Wikidata*, which contains also not that many unique predicates, we analyzed around 35K URIs. Note that predicates which are derived from relations using a suffix (e.g., the suffix "s" as in `wdt:P1024s` is used for predicates referring to a statement) could not be dereferenced at all. Furthermore, the blank nodes used for reification cannot be dereferenced.

Regarding *Freebase*, mainly all URIs on subject and object position of triples could be dereferenced. Some resources were not resolvable even after multiple attempts (HTTP server error 503; e.g., `freebase:m.0156q`). Surprisingly, server errors also appeared while browsing the website freebase.com, so that data was partially not available. Regarding the predicate position, many URIs are not dereferencable due to server errors (HTTP 503) or due to unknown URIs (HTTP 404). Note that if a large number of Freebase requests are performed, an API key from Google is necessary. In our experiments, the access was blocked after a few thousand requests. Hence, we can point out that without an API key the Freebase KG is only usable to a limited extent.

Availability of the KG, m_{Avai}

Evaluation method. We measured the availability of the officially hosted KGs with the monitoring service Pingdom.[122] For each KG, an uptime test was set up, which checked the availability of the resource "Hamburg" as representative resource for successful URI resolving (i.e., returning the status code HTTP 200) every minute over the time range of 60 days (Dec 18, 2015–Feb 15, 2016).

Evaluation result. While the other KGs showed almost no outages and were again online after some minutes on average, YAGO outages took place frequently and lasted on average 3.5 hours.[123] In the given time range, four outages took longer than one day. Based on these insights, we recommend to use a local version of YAGO for time-critical queries.

Availability of a public SPARQL endpoint, m_{SPARQL}

The SPARQL endpoints of DBpedia and YAGO are provided by a Virtuoso server,[124] the Wikidata SPARQL endpoint via Blazegraph.[125] Freebase and OpenCyc do not provide an official SPARQL endpoint. However, an endpoint for the MQL query language for the Freebase KG was available.

Especially regarding the Wikidata SPARQL endpoint we observed access restrictions: The maximum execution time per query is set to 30 seconds, but there is no limitation regarding the returning number of rows. However, the front-end of the SPARQL endpoint crashed in case of large result sets with more than 1.5M rows. Although public SPARQL endpoints need to be prepared for inefficient queries, the time limit of Wikidata may impede the execution of reasonable queries.

Provisioning of an RDF export, m_{Export}

All considered KGs provide RDF exports as downloadable files. The format of the data differs from KG to KG. Mostly, data is provided in N-Triples and Turtle format.

Support of content negotiation, m_{Negot}

We measure the support of content negotiation regarding the serialization formats RDF/XML, N3/Turtle, and N-Triples. OpenCyc does not provide any content negotiation; only RDF/XML is supported as content type. Therefore, OpenCyc does not fulfill the criterion of supporting content negotiation.

The endpoints for DBpedia, Wikidata, and YAGO correctly returned the appropriate RDF serialization format and the corresponding HTML representation of the tested resources. Freebase does currently not provide any content negotiation and only the content type `text/plain` is returned.

Noteworthy is also that regarding the N-Triples serialization YAGO and DBpedia require the accept header `text/plain` and not `applicationn-triples`. This is due to the usage of Virtuoso as endpoint. For DBpedia, the forwarding to `http://dbpedia.org/data/[resource].ntriples` does not work; instead, the HTML representation is returned. Therefore, the KG is evaluated with 0.5.

Linking HTML sites to RDF serializations, $m_{HTMLRDF}$

All KGs except OpenCyc interlink the HTML representations of resources with the corresponding RDF representations by means of `<link rel="alternate"`

[122] See `https://www.pingdom.com`, requested Mar 2, 2016. The HTTP requests of Pingdom are executed by various servers so that caching is prevented.

[123] See diagrams per KG on our website (`http://km.aifb.kit.edu/sites/knowledge-graph-comparison/`; requested on Jan 31, 2017).

[124] See `https://virtuoso.openlinksw.com/`, requested on Dec 28, 2016.

[125] See `https://www.blazegraph.com/`, requested on Dec 28, 2016.

Table 13
Evaluation results for the KGs regarding the dimension *License*.

	DB	FB	OC	WD	YA
$m_{macLicense}$	1	0	0	1	0

`type="{content type}" href="{URL}"/>` in the HTML header.

Provisioning of metadata about the KG, m_{meta}

For this criterion we analyzed if KG metadata is available, such as in the form of a VoID file.[126] DBpedia integrates the VoID vocabulary directly in its KG[127] and provides information such as the SPARQL endpoint URL and the number of all triples. OpenCyc reveals the current KG version number via `owl:versionInfo`. For YAGO, Freebase, and Wikidata no meta information could be found.

5.2.10. License

The evaluation results of the dimension *License* are shown in Table 13.

Provisioning machine-readable licensing information, $m_{macLicense}$

DBpedia and Wikidata provide licensing information about their KG data in machine-readable form. For DBpedia, this is done in the ontology via the predicate `cc:license` linking to *CC-BY-SA*[128] and *GNU Free Documentation License* (GNU FDL).[129] Wikidata embeds licensing information during the dereferencing of resources in the RDF document by linking with `cc:license` to the license *CC0*.[130] YAGO and Freebase do not provide machine-readable licensing information. However, their data is published under the license *CC-BY*.[131] OpenCyc embeds licensing information into the RDF document during dereferencing, but not in machine-readable form.[132]

[126] See `https://www.w3.org/TR/void/`, requested on Apr 7, 2016.

[127] See `http://dbpedia.org/void/page/Dataset`, requested on Mar 5, 2016.

[128] See `http://creativecomons.org/licenses/by-sa/3.0/`, requested on Feb 4, 2017.

[129] See `http://www.gnu.org/copyleft/fdl.html`, requested on Feb 4, 2017.

[130] See `http://creativecomons.org/publicdomain/zero/1.0/`, requested on Feb 4, 2017.

[131] See `http://createivecommons.org/licenses/by/3.0`, requested on Feb 4, 2017.

[132] License information is provided as plain text among further information with the relation `rdfs:comment`.

Table 14
Evaluation results for the KGs regarding the dimension *Interlinking*.

	DB	FB	OC	WD	YA
m_{Inst}	0.25	0	0.38	0 (.09)	0.31
m_{URIs}	0.93	0.91	0.89	0.96	0.96

5.2.11. Interlinking

The evaluation results of the dimension *Interlinking* are shown in Table 14.

Linking via owl:sameAs, m_{Inst}

Evaluation method. Given all `owl:sameAs` triples in each KG, we queried all those subjects thereof which are instances but neither classes nor relations[133] and where the resource in the object position of the triple is an external source, i.e., not belonging to the namespace of the KG.

Evaluation result. OpenCyc and YAGO achieve the best results w.r.t. this metric, but DBpedia has by far the most instances with at least one `owl:sameAs` link. We can therefore confirm the statement by Bizer et al. [12] that DBpedia has established itself as a hub in the Linked Data cloud.

In *DBpedia*, there are about 5.2M instances with at least one `owl:sameAs` link. Links to localized DBpedia versions (e.g., `de.dbpedia.org`) were counted as internal links and, hence, not considered here. In total, one-fourth of all instances have at least one `owl:sameAs` link.

In *Wikidata*, neither `owl:sameAs` links are provided nor a corresponding proprietary relation is available. Instead, Wikidata uses for each linked data set a proprietary relation (called "identifier") to indicate equivalence. For example, the M-ID of a Freebase instance is stored via the relation “Freebase identifier” (`wdt:P646`) as literal value (e.g., `"/m/01x3gpk"`). So far, links to 426 different data sources are maintained in this way.

Although the equivalence statements in Wikidata can be used to generate corresponding `owl:sameAs` statements and although the stored identifiers are provided in the Browser interface as hyperlinks, there are no genuine `owl:sameAs` links available. Hence, Wikidata is evaluated with 0. If we view each equivalence relation as `owl:sameAs` relation, we would obtain around 12.2M instances with `owl:sameAs` statements. This corresponds to 8.6% of all instances. If we consider

[133] The interlinking on schema level is already covered by the criterion *Interoperability of proprietary vocabulary*.

only entities instead of instances (since there are many instances due to reification), we obtain a coverage of 65%. Note, however, that, although the linked resources provide relevant content, the resources are not always RDF documents, but instead HTML web pages. Therefore, we cannot easily subsume all "identifiers" (equivalence statements) under `owl:sameAs`.

YAGO has around 3.6M instances with at least one `owl:sameAs` link. However, most of them are links to DBpedia based on common Wikipedia articles. If those links are excluded, YAGO contains mostly links to GeoNames and would be evaluated with just 0.01.

In case of *OpenCyc*, links to Cyc,[134] the commercial version of OpenCyc, were considered as being internal. Still, OpenCyc has the highest fulfillment degree with around 40K instances with at least one `owl:sameAs` link. As mentioned earlier, the relative high linking degree of OpenCyc can be attributed to dedicated approaches of linking OpenCyc to other KGs.[135]

Validity of external URIs, m_{URIs}

Regarding the dimension *Accessibility*, we already analyzed the dereferencing possibility of resources in the KG namespace. Now we analyze the links to external URIs.

Evaluation method. External links include `owl:sameAs` links as well as links to non-RDF-based Web resources (e.g., via `foaf:homepage`). We measure errors such as timouts, client errors (HTTP response 4xx), and server errors (HTTP response 5xx).

Evaluation result. The external links are in most of the cases valid for all KGs. All KGs obtain a metric value between 0.89 and 0.96.

DBpedia stores provenance information via the relation `prov:wasDerivedFrom`. Since almost all links refer to Wikipedia, 99% of the resources are available.

Freebase achieves high metric values here, since it contains `owl:sameAs` links mainly to Wikipedia. Also Wikipedia URIs are mostly resolvable.

OpenCyc contains mainly external links to non-RDF-based Web resources to wikipedia.org and w3.org.

YAGO also achieves high metric values, since it provides `owl:sameAs` links only to DBpedia and GeoNames, whose URIs do not change.

For *Wikidata* the relation "reference URL" (`wdt:P854`), which states provenance information among other relations, belongs to the links linking to external Web resources. Here we were able to resolve around 95.5% without errors.

Noticeable is that DBpedia and OpenCyc contain many `owl:sameAs` links to URIs whose domains do not exist anymore.[136] One solution for such invalid links might be to remove them if they have been invalid for a certain time span.

5.2.12. Summary of Results

We now summarize the results of the evaluations presented in this section.

1. *Syntactic validity of RDF documents:* All KGs provide syntactically valid RDF documents.
2. *Syntactic validity of Literals:* In general, the KGs achieve good scores regarding the *Syntactic validity of literals*. Although OpenCyc comprises over 1M literals in total, these literals are mainly labels and descriptions which are not formatted in a special format. For YAGO, we detected about 519K syntactic errors (given 1M literal values) due to the usage of wildcards in the date values. Obviously, the syntactic invalidity of literals is accepted by the publishers in order to keep the number of relations low. In case of Wikidata, some invalid literals such as the ISBN have been corrected in newer versions of Wikidata. This indicates that knowledge in Wikidata is curated continuously. For DBpedia, comments next to the values to be extracted (such as ISBN) in the infoboxes of Wikipedia led to inaccurately extracted values.
3. *Semantic validity of triples:* All considered KGs scored well regarding this metric. This shows that KGs can be used in general without concerns regarding the correctness. Note, however, that evaluating the semantic validity of facts is very challenging, since a reliable ground truth is needed.
4. *Trustworthiness on KG level:* Based on the way of how data is imported and curated, OpenCyc and Wikidata can be trusted the most.
5. *Trustworthiness on statement level:* Here, especially good values are achieved for Freebase, Wikidata, and YAGO. YAGO stores per statement both the source and the extraction technique, which is unique among the KGs. Wikidata also supports to store the source of information, but only around 1.3% of the statements have provenance information attached. Note, however, that not every state-

[134] I.e., `sw.cyc.com`

[135] See *Interoperability of proprietary vocabulary* in sec. 5.2.8.

[136] E.g., `http://rdfabout.com`, `http://www4.wiwiss.fu-berlin.de/factbook/`, and `http://wikicompany.org` (requested on Jan 11, 2017).

ment in Wikidata requires a reference and that it is hard to evaluate which statements lack such a reference.

6. *Using unknown and empty values:* Wikidata and Freebase support the indication of unknown and empty values.
7. *Check of schema restrictions during insertion of new statements:* Since Freebase and Wikidata are editable by community members, simple consistency checks are made during the insertion of new facts in the user interface.
8. *Consistency of statements w.r.t. class constraints:* Freebase and Wikidata do not specify any class constraints via `owl:disjointWith`, while the other KGs do.
9. *Consistency of statements w.r.t. relation constraints:* The inconsistencies of all KGs regarding the range indications of relations are mainly due to inconsistently used data types (e.g., `xsd:gYear` is used instead of `xsd:Date`).
   Regarding the constraint of functional properties, the relation `owl:FunctionalProperty` is used by all KGs except Wikidata; in most cases the KGs comply with the usage restrictions of this relation.
10. *Creating a ranking of statements:* Only Wikidata supports a ranking of statements. This is in particular worthwhile in case of statements which are only temporally limited valid.
11. *Schema completeness:* Wikidata shows the highest degree of schema completeness. Also for DBpedia, OpenCyc, and YAGO we obtain results which are presumably acceptable in most cross-domain use cases. While DBpedia classes were sometimes missing in our evaluation, the DBpedia relations were covered considerably well. OpenCyc lacks some relations of the gold standard, but the classes of the gold standard were existing in OpenCyc. While the YAGO classes are peculiar in the sense that they are connected to WordNet synsets, it is remarkable that YAGO relations are often kept very abstract so that they can be applied in different senses. Freebase shows considerable room for improvement concerning the coverage of typical cross-domain classes and relations. Note that Freebase classes are belonging to different domains. Hence, it is difficult to find related classes if they are not in the same domain.
12. *Column completeness:* DBpedia and Freebase show the best column completeness values, i.e., in those KGs the predicates used by the instances of each class are on average frequently used by all of those class instances. We can name data imports as one reason for it.
13. *Population completeness:* Not very surprising is the fact that all KGs show a higher degree of completeness regarding well-known entities than regarding rather unknown entities. Especially Wikidata shows an excellent performance for both well-known and rather unknown entities.
14. *Timeliness frequency of the KG:* Only Wikidata provides the highest fulfillment degree for this criterion, as it is continuously updated and as the changes are immediately visible and queryable by users.
15. *Specification of the validity period of statements:* In YAGO, Freebase, and Wikidata the temporal validity period of statements (e.g., term of office) can be specified.
16. *Specification of the modification date of statements:* Only Freebase keeps the modification dates of statements. Wikidata provides the modification date of the queried resource during URI dereferencing.
17. *Description of resources:* YAGO, Wikidata, and OpenCyc contain a label for almost every entity. Surprisingly, DBpedia shows a relatively low coverage w.r.t. labels and descriptions (only 70.4%). Manual investigations suggest that the intermediate node mapping template is the main reason for that. By means of this template, intermediate nodes are introduced and instantiated, but no labels are provided for them.[137]
18. *Labels in multiple languages:* YAGO, Freebase, and Wikidata support hundreds of languages regarding their stored labels. Only OpenCyc contains labels merely in English. While DBpedia, YAGO, and Freebase show a high coverage regarding the English language, Wikidata does not have such a high coverage regarding English, but instead covers other languages to a considerable extent. It is, hence, not only the most diverse KG in terms of languages, but also the KG which contains the most labels for languages other than English.
19. *Understandable RDF serialization:* DBpedia, Wikidata, and YAGO provide several understand-

[137] An example is `dbr:Volkswagen_Passat_(B1)`, which has `dbo:engine` statements to the intermediate nodes `Volkswagen_Passat_(B1)__1`, etc., representing different engine variations.

able RDF serialization formats. Freebase only provides the understandable format RDF/Turtle. OpenCyc relies only on RDF/XML, which is considered as being not easily understandable for humans.

20. *Self-describing URIs:* We can find mixed paradigms regarding the URI generation: DBpedia, YAGO, and OpenCyc rely on descriptive URIs, while Wikidata and Freebase (in part; classes and relations are identified with self-describing URIs) use generic IDs, i.e., opaque URIs.
21. *Avoiding blank nodes and RDF reification:* DBpedia, Wikidata, YAGO, and Freebase are the KGs which use reification, i.e., which formulate statements about statements. There are different ways of implementing reification [27]. DBpedia, Wikidata, and Freebase use *n-ary relations*, while YAGO uses *N-Quads*, creating so-called *named graphs*.
22. *Provisioning of several serialization formats:* Many KGs provide RDF in several serialization formats. Freebase is the only KG providing data in the serialization format RDF/Turtle only.
23. *Using external vocabulary:* DBpedia and Wikidata show high degrees of external vocabulary usage. In DBpedia the RDF, RDFS, and OWL vocabularies are used. Wikidata has a high external vocabulary ratio, since there exist many language labels and descriptions (modeled via `rdfs:label` and `schema:description`). Also, due to instantiations of statements with `wdo:Statement` for reification purposes, the external relation `rdf:type` is used a lot.
24. *Interoperability of proprietary vocabulary:* We obtained low fulfillment scores regarding this criterion. OpenCyc shows the highest value. We can mention as reason for that the fact that half of all OpenCyc classes exhibit at least one `owl:sameAs` link.
    While DBpedia has equivalence statements to external classes for almost every second class, only 6.3% of all relations have equivalence relations to relations outside the DBpedia namespace.
    Wikidata shows a very low interlinking degree of classes to external classes and of relations to external relations.
25. *Dereferencing possibility of resources:* Resources in DBpedia, OpenCyc, and YAGO can be dereferenced without considerable issues. Wikidata uses predicates derived from relations that are not dereferencable at all, as well as blank nodes. For Freebase we measured a quite considerable amount of dereferencing failures due to server errors and unknown URIs. Note also that Freebase required an API key for a large amount of requests.
26. *Availability of the KG:* While all other KGs showed almost no outages, YAGO shows a noteworthy instability regarding its online availability. We measured around 100 outages for YAGO in a time interval of 8 weeks, taking on average 3.5 hours.
27. *Provisioning of public SPARQL endpoint:* DBpedia, Wikidata, and YAGO provide a SPARQL endpoint, while Freebase and OpenCyc do not. Noteworthy is that the Wikidata SPARQL endpoint has a maximum execution time per query of 30 seconds. This might be a bottleneck for some queries.
28. *Provisioning of an RDF export:* RDF exports are available for all KGs and are provided mostly in N-Triples and Turtle format.
29. *Support of content negotiation:* DBpedia, Wikidata, and YAGO correctly return RDF data based on content negotiation. Both OpenCyc and Freebase do not support any content negotiation. While OpenCyc only provides data in RDF/XML, Freebase only returns data with `text/plain` as content type.
30. *Linking HTML sites to RDF serializations:* All KGs except OpenCyc interlink the HTML representations of resources with the corresponding RDF representations.
31. *Provisioning of KG metadata:* Only DBpedia and OpenCyc integrate metadata about the KG in some form. DBpedia has the VoID vocabulary integrated, while OpenCyc reveals the current KG version as machine-readable metadata.
32. *Provisioning machine-readable licensing information:* Only DBpedia and Wikidata provide licensing information about their KG data in machine-readable form.
33. *Interlinking via `owl:sameAs`:* OpenCyc and YAGO achieve the best results w.r.t. this metric, but DBpedia has by far the most instances with at least one `owl:sameAs` link. Based on the resource interlinkage, DBpedia is justifiably called Linked Data hub. Wikidata does not provide `owl:sameAs` links but stores identifiers as literals that could be used to generate `owl:sameAs` links.
34. *Validity of external URIs:* The links to external Web resources are for all KGs valid in most cases. DBpedia and OpenCyc contain many

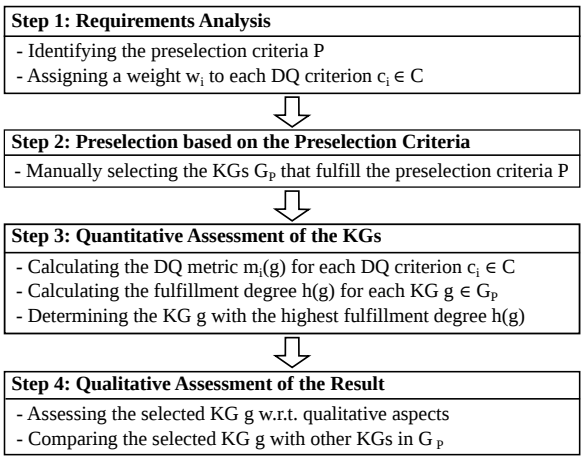

Fig. 11. Proposed process for using our KG recommendation framework.

`owl:sameAs` links to RDF documents on domains which do not exist anymore; those links could be deleted.

6. KG Recommendation Framework

We now propose a framework for selecting the most suitable KG (or a set of suitable KGs) for a given concrete setting, based on a given set of KGs $G = \{g_1, ..., g_n\}$. To use this framework, the user needs to go through the steps depicted in Fig. 11.

In Step 1, the preselection criteria and the weights for the criteria are specified. The preselection criteria can be both quality criteria or general criteria and need to be selected dependent on the use case. The *Timeliness frequency of the KG* is an example for a quality criterion. The license under which a KG is provided (e.g., CC0 license) is an example for a general criterion. After weighting the criteria, in Step 2 those KGs are neglected which do not fulfill the preselection criteria. In Step 3, the fulfillment degrees of the remaining KGs are calculated and the KG with the highest fulfillment degree is selected. Finally, in Step 4 the result can be assessed w.r.t. qualitative aspects (besides the quantitative assessments using the DQ metrics) and, if necessary, an alternative KG can be selected for being applied for the given scenario.

Use case application. In the following, we show how to use the *KG recommendation framework* in a particular scenario. The use case is based on the usage of DBpedia and MusicBrainz for the project *BBC Music* as described in [33].

Description of the use case: The publisher BBC wants to enrich news articles with fact sheets providing relevant information about musicians mentioned in the articles. In order to obtain more details about the musicians, the user can leave the news section and access the musicians section where detailed information is provided, including a short description, a picture, the birth date, and the complete discography for each musician. For being able to integrate the musicians information into the articles and to enable such a linking, editors shall tag the article based on a controlled vocabulary.

The *KG Recommendation Framework* can be applied as follows:

1. *Requirements analysis*:
   - *Preselection criteria*: According to the scenario description [33], the KG in question should (i) be actively curated and (ii) contain an appropriate amount of media entities. Given these two criteria, a satisfactory and up-to-date coverage of both old and new musicians is expected.
   - *Weighting of DQ criteria*: Based on the preselection criteria, an example weighting of the DQ metrics for our use case is given in Table 15. Note that this is only one example configuration and the assignment of the weights is subjective to some degree. Given the preselection criteria, the criterion *Timeliness frequency of the KG* and the criteria of the DQ dimension *Completeness* are emphasized. Furthermore, the criteria *Dereferencing possibility of resources* and *Availability of the KG* are important, as the KG shall be available online, ready to be queried.[138]
2. *Preselection*: Freebase and OpenCyc are not considered any further, since Freebase is not being updated anymore and since OpenCyc contains only around 4K entities in the media domain.
3. *Quantitative Assessment*: The overall fulfillment score for each KG is calculated based on the formula presented in Section 3.1. The result of the quantitative KG evaluation is presented in Table 15. By weighting the criteria according to the constraints, Wikidata achieves the best rank, closely followed by DBpedia. Based on the quantitative assessment, Wikidata is recommended by the framework.

[138] We assume that in this use case rather the dereferencing of HTTP URIs than the execution of SPARQL queries is desired.

Table 15
Framework with an example weighting which would be reasonable for a user setting as given in [33].

Dimension	**Metric**	**DBpedia**	**Freebase**	**OpenCyc**	**Wikidata**	**YAGO**	**Example of User Weighting w_i**
Accuracy	m_{synRDF}	1	1	1	1	1	1
	m_{synLit}	0.994	1	1	1	0.624	1
	$m_{semTriple}$	0.990	0.995	1	0.993	0.993	1
Trustworthiness	m_{graph}	0.5	0.5	1	0.75	0.25	0
	m_{fact}	0.5	1	0	1	1	1
	m_{NoVal}	0	1	0	1	0	0
Consistency	$m_{checkRestr}$	0	1	0	1	0	0
	$m_{conClass}$	0.875	1	0.999	1	0.333	0
	$m_{conRelat}$	0.992	0.451	1	0.500	0.992	0
Relevancy	$m_{Ranking}$	0	1	0	1	0	1
Completeness	$m_{cSchema}$	0.905	0.762	0.921	1	0.952	1
	m_{cCol}	0.402	0.425	0	0.285	0.332	2
	m_{cPop}	0.93	0.94	0.48	0.99	0.89	3
Timeliness	m_{Freq}	0.5	0	0.25	1	0.25	3
	$m_{Validity}$	0	1	0	1	1	0
	m_{Change}	0	1	0	0	0	0
Ease of understanding	m_{Descr}	0.704	0.972	1	0.9999	1	1
	m_{Lang}	1	1	0	1	1	0
	m_{uSer}	1	1	0	1	1	0
	m_{uURI}	1	0.5	1	0	1	1
Interoperability	m_{Reif}	0.5	0.5	0.5	0	0.5	0
	$m_{iSerial}$	1	0	0.5	1	1	1
	m_{extVoc}	0.61	0.108	0.415	0.682	0.134	1
	$m_{propVoc}$	0.150	0	0.513	0.001	0	1
Accessibility	m_{Deref}	1	0.437	1	0.414	1	2
	m_{Avai}	0.9961	0.9998	1	0.9999	0.7306	2
	m_{SPARQL}	1	0	0	1	1	1
	m_{Export}	1	1	1	1	1	0
	m_{Negot}	0.5	0	0	1	1	0
	$m_{HTMLRDF}$	1	1	0	1	1	0
	m_{Meta}	1	0	1	0	0	0
Licensing	$m_{macLicense}$	1	0	0	1	0	0
Interlinking	m_{Inst}	0.251	0	0.382	0	0.310	3
	m_{URIs}	0.929	0.908	0.894	0.957	0.956	1
Unweighted Average		0.683	0.603	0.496	0.752	0.625	
Weighted Average		0.701	0.493	0.556	0.714	0.648	

4. *Qualitative Assessment*: The high population completeness in general and the high coverage of entities in the media domain in particular give Wikidata advantage over the other KGs. Furthermore, Wikidata does not require that there is a Wikipedia article for each entity. Thus, missing Wikidata entities can be added by the editors directly and are then available immediately.
   The use case requires to retrieve also detailed information about the musicians from the KG, such as a short descripion and a discography. DBpedia tends to store more of that data, especially w.r.t. discography. A specialized database like MusicBrainz provides even more data about musicians than DBpedia, as it is not limited to the Wikipedia infoboxes. While DBpedia does not provide any links to MusicBrainz, Wikidata stores around 120K equivalence links to MusicBrainz that can be used to pull more data. In conclusion, Wikidata, especially in the combination with MusicBrainz, seems to be an appropriate choice for the use case. In this case, the qualitative assessment confirms the result of the quantitative assessment.

The use case shows that our *KG recommendation framework* enables users to find the most suitable KG and is especially useful in giving an overview of the most relevant criteria when choosing a KG. However, applying our framework to the use case also showed that, besides the quantitative assessment, there is still a need for a deep understanding of commonalities and difference of the KGs in order to make an informed choice.

7. Related Work

7.1. Linked Data Quality Criteria

Zaveri et al. [49] provide a conceptual framework for quality assessment of linked data based on quality criteria and metrics which are grouped into quality dimensions and categories and which are based on the framework of Wang et al. [47]. Our framework is also based on Wang's dimensions and extended by the dimensions *Consistency* [11], *Licensing* and *Interlinking* [49]. Furthermore, we reintroduce the dimensions *Trustworthiness* and *Interoperability* as a collective term for multiple dimensions.

Many published DQ criteria and metrics are rather abstract. We, in contrast, selected and developed concrete criteria which can be applied to any KG in the Linked Open Data cloud. Table 16 shows which of the metrics introduced in this article have already been used to some extent in existing literature. In summary, related work mainly proposed generic guidelines for publishing Linked Data [26], introduced DQ criteria with corresponding metrics (e.g., [20,30]) and criteria without metrics (e.g., [40,29]). 27 of the 34 criteria introduced in this article have been introduced or supported in one way or another in earlier works. The remaining seven criteria, namely *Trustworthiness on KG level*, m_{graph}, *Indicating unknown and empty values*, m_{NoVal}, *Check of schema restrictions during insertion of new statements*, $m_{checkRestr}$, *Creating a ranking of statements*, $m_{Ranking}$, *Timeliness frequency of the KG*, m_{Freq}, *Specification of the validity period of statements*, $m_{Validity}$, and *Availability of the KG*, m_{Avai}, have not been proposed so far, to the best of our knowledge. In the following, we present more details of single existing approaches for Linked Data quality criteria.

Pipino et al. [40] introduce the criteria *Schema completeness*, *Column completeness* and *Population completeness* in the context of databases. We introduce those metrics for KGs and apply them, to the best of our knowledge, the first time on the KGs DBpedia, Freebase, OpenCyc, Wikidata, and YAGO.

OntoQA [45] introduces criteria and corresponding metrics that can be used for the analysis of ontologies. Besides simple statistical figures such as the average of instances per class, Tartir et al. introduce also criteria and metrics similar to our DQ criteria *Description of resources*, m_{Descr}, and *Column completeness*, m_{cCol}.

Based on a large-scale crawl of RDF data, Hogan et al. [29] analyze quality issues of published RDF data. Later, Hogan et al. [30] introduce further criteria and metrics based on Linked Data guidelines for data publishers [26]. Whereas Hogan et al. crawl and analyze many KGs we analyze a selected set of KGs in more detail.

Heath et al. [26] provide guidelines for Linked Data but do not introduce criteria or metrics for the assessment of Linked Data quality. Still, the guidelines can be easily translated into relevant criteria and metrics. For instance, "Do you refer to additional access methods" leads to the criteria *Provisioning of public SPARQL endpoint*, m_{SPARQL}, and *Provisioning of an RDF export*, m_{Export}. Also, "Do you map proprietary vocabulary terms to other vocabularies?" leads to the criterion *Interoperability of proprietary vocabulary*, $m_{propVoc}$. Metrics that are based on the guidelines of Heath et al. can also be found in other frameworks [30,20].

Table 16
Overview of related work regarding data quality criteria for KGs.

DQ Metric	[40]	[45]	[29]	[26]	[20]	[22]	[30]	[48]	[2]	[34]
m_{synRDF}			✓		✓					
m_{synLit}			✓			✓		✓		✓
$m_{semTriple}$						✓		✓	✓	✓
m_{fact}				✓	✓					
$m_{conClass}$			✓		✓					✓
$m_{conRelat}$			✓		✓	✓		✓	✓	✓
$m_{cSchema}$	✓					✓				
m_{cCol}	✓	✓				✓				✓
m_{cPop}	✓					✓				
m_{Change}					✓	✓				
m_{Descr}		✓		✓	✓		✓			
m_{Lang}					✓					
m_{uSer}				✓						
m_{uURI}							✓			
m_{Reif}			✓	✓			✓			
$m_{iSerial}$					✓					
m_{extVoc}				✓			✓			
$m_{propVoc}$				✓						
m_{Deref}			✓	✓	✓		✓			
m_{SPARQL}				✓						
m_{Export}				✓	✓					
m_{Negot}			✓	✓	✓					
$m_{HTMLRDF}$				✓						
m_{Meta}				✓	✓		✓			
$m_{macLicense}$			✓	✓		✓				
m_{Inst}			✓			✓	✓			
m_{URIs}					✓			✓		

Flemming [20] introduces a framework for the quality assessment of Linked Data quality. This framework measures the Linked Data quality based on a sample of a few RDF documents. Based on a systematic literature review, criteria and metrics are introduced. Flemming introduces the criteria *Labels in multiple languages*, m_{Lang}, and *Validity of external URIs*, m_{URIs}, the first time. The framework is evaluated on a sample of RDF documents of DBpedia. In contrast to Flemming, we evaluate the whole KG DBpedia and also four other widely used KGs.

SWIQA[22] is a quality assessment framework introduced by Fürber et al. that introduces criteria and metrics for the dimensions *Accuracy*, *Completeness*, *Timeliness*, and *Uniqueness*. In this framework, the dimension *Accuracy* is divided into *Syntactic validity* and *Sematic validity* as proposed by Batini et al. [6]. Furthermore, the dimension *Completeness* comprises *Schema completeness*, *Column completeness* and *Population completeness*, following Pipino et al. [40]. In this article, we make the same distinction, but in addition distinguish between RDF documents, RDF triples, and RDF

literals for evaluating the *Accuracy*, since we consider RDF KGs.

TripleCheckMate [35] is a framework for Linked Data quality assessment using a crowdsourcing-approach for the manual validation of facts. Based on this approach, Zaveri et al. [48] and Acosta et al. [2,3] analyze both syntactic and semantic accuracy as well as the consistency of data in DBpedia.

Kontokostas et al. [34] present the test-driven evaluation framework RDFUnit for assessing Linked Data quality. This framework is inspired by the paradigm of test-driven software development. The framework introduces 17 SPARQL templates of tests that can be used for analyzing KGs w.r.t. *Accuracy* and *Consistency*. Note that those tests can also be used for evaluating external constraints that exist due to the usage of external vocabulary. The framework is applied by Kontokostas et al. on a set of KGs including DBpedia.

7.2. Comparing KGs by Key Statistics

Duan et al. [14], Tartir [45], and Hassanzadeh [25] can be mentioned as the most similar related work regarding the evaluation of KGs using the key statistics presented in Section 5.1.

Duan et al. [14] analyze the structuredness of data in DBpedia, YAGO2, UniProt, and in several benchmark data sets. To that end, the authors use simple statistical key figures that are calculated based on the corresponding RDF dumps. In contrast to that approach, we use SPARQL queries to obtain the figures, thus not limiting ourselves to the N-Tripel serialization of RDF dump files. Duan et al. claim that simple statistical figures are not sufficient to gain fruitful findings when analyzing the structuredness and differences of RDF datasets. The authors therefore propose in addition a coherence metric. Accordingly, we analyze not only simple statistical key figures but further analyze the KGs w.r.t. data quality, using 34 DQ metrics.

Tartir et al. [45] introduce with the system OntoQA metrics that can be used for analyzing ontologies. More precisely, it can be measured to which degree the schema level information is actually used on instance level. An example of such a metric is the class richness, defined as the number of classes with instances divided by the number of classes without instances. SWETO, TAP, and GlycO are used as showcase ontologies.

Tartir et al. [45] and Hassanzadeh et al. [25] analyze how domains are covered by KGs on both schema and instance level. For that, Tartir et al. introduce the measure *importance* as the number of instances per class and their subclasses. In our case, we cannot use this approach, since Freebase has no hierarchy. Hassanzadeh et al. analyze the coverage of domains by listing the most frequent classes with the highest number of instances as a table. This gives only little overview of the covered domains, since instances can belong to multiple classes in the same domain, such as `dbo:Place` and `dbo:PopulatedPlace`. For determining the domain coverages of KGs for this article, we therefore adapt the idea of Hassanzadeh et al. by manually mapping the most frequent classes to domains and deleting duplicates within the domains. That means, if an instance is instantiated both as `dbo:Place` and `dbo:PopulatedPlace`, the instance will be counted only once in the domain geography.

8. Conclusion

Freely available knowledge graphs (KGs) have not been in the focus of any extensive comparative study so far. In this survey, we defined a range of aspects according to which KGs can be analyzed. We analyzed and compared DBpedia, Freebase, OpenCyc, Wikidata, and YAGO along these aspects and proposed a framework as well as a process to enable readers to find the most suitable KG for their settings.

References

[1] M. Acosta, E. Simperl, F. Flöck, and M. Vidal. HARE: A Hybrid SPARQL Engine to Enhance Query Answers via Crowdsourcing. In *Proceedings of the 8th International Conference on Knowledge Capture*, K-CAP 2015, pages 11:1–11:8. ACM, 2015.

[2] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann. Crowdsourcing linked data quality assessment. In *The Semantic Web–ISWC 2013*, pages 260–276. Springer, 2013.

[3] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, F. Flöck, and J. Lehmann. Detecting Linked Data Quality Issues via Crowdsourcing: A DBpedia Study. *Semantic Web*, 2016.

[4] 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 and 2nd Asian Semantic Web Conference*, ISWC 2007/ASWC 2007, pages 722–735. Springer, 2007.

[5] S. Auer, J. Lehmann, A.-C. Ngonga Ngomo, and A. Zaveri. Introduction to Linked Data and Its Lifecycle on the Web. In *Reasoning Web. Semantic Technologies for Intelligent Data Access*, volume 8067 of *Lecture Notes in Computer Science*, pages 1–90. Springer Berlin Heidelberg, 2013.

[6] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for Data Quality Assessment and Improvement. *ACM Comput. Surv.*, 41(3):16:1–16:52, July 2009.

[7] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. OWL Web Ontology Language Reference. https://www.w3.org/TR/2004/REC-owl-ref-20040210, 2004. [Online; accessed 06-Apr-2016].

[8] T. Berners-Lee. Linked Data. http://www.w3.org/DesignIssues/LinkedData.html, 2006. [Online; accessed 28-Feb-2016].

[9] T. Berners-Lee. Linked Data Is Merely More Data. http://www.w3.org/DesignIssues/LinkedData.html, 2006. [Online; accessed 28-02-2016].

[10] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):29–37, 5 2001.

[11] C. Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. VDM Publishing, 2007.

[12] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia–A crystallization point for the Web of Data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[13] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, New York, NY, USA, 2014. ACM.

[14] S. Duan, A. Kementsietsidis, K. Srinivas, and O. Udrea. Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD 2011, pages 145–156, 2011.

[15] B. Ell, D. Vrandečić, and E. Simperl. *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, chapter Labels in the Web of Data, pages 162–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[16] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandecic. Introducing Wikidata to the Linked Data Web. In *Proceedings of the 13th International Semantic Web Conference*, ISWC 2014, pages 50–65. Springer, 2014.

[17] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger. Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*, 2017, to be published.

[18] M. Färber, C. Menne, and A. Rettinger. A Linked Data Wrapper for CrunchBase. *Semantic Web Journal*, 2017, to be published.

[19] C. Fellbaum. *WordNet – An Electronic Lexical Database*. MIT Press, 1998.

[20] A. Flemming. Qualitätsmerkmale von Linked Data-veröffentlichenden Datenquellen (Quality characteristics of linked data publishing datasources). *Diploma Thesis, Humboldt University of Berlin,* http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/diploma_seminar_thesis/Diplomarbeit_Annika_Flemming.pdf, 2011.

[21] G. Freedman and E. G. Reynolds. Enriching Basal Reader Lessons with Semantic Webbing. *Reading Teacher*, 33(6):677–684, 1980.

[22] C. Fürber and M. Hepp. SWIQA – A Semantic Web Information Quality Assessment Framework. In *Proceedings of the 19th European Conference on Information Systems (ECIS2011)*, volume 15, page 19, 2011.

[23] R. Guns. Tracing the origins of the Semantic Web. *Journal of the American Society for Information Science and Technology*, 64(10):2173–2181, 2013.

[24] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson. *The Semantic Web – ISWC 2010: 9th International Semantic Web Conference, ISWC 2010, Shanghai, China*, chapter When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data, pages 305–320. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[25] O. Hassanzadeh, M. J. Ward, M. Rodriguez-Muro, and K. Srinivas. Understanding a Large Corpus of Web Tables Through Matching with Knowledge Bases – An Empirical Study. In *Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference, ISWC 2015*, 2015.

[26] T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[27] D. Hernández, A. Hogan, and M. Krötzsch. Reifying RDF: What Works Well With Wikidata? In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference*, pages 32–47, 2015.

[28] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[29] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the Pedantic Web. *Proceedings of the WWW2010 Workshop on Linked Data on the Web*, 628, 2010.

[30] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of linked data conformance. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:14–44, 2012.

[31] P. Jain, P. Hitzler, K. Janowicz, and C. Venkatramani. There's No Money in Linked Data. http://corescholar.libraries.wright.edu/cse/240, 2013. accessed July 20, 2015.

[32] J. M. Juran, F. M. Gryna, and R. S. Bingham, editors. *Quality Control Handbook*. McGraw-Hill, 1974.

[33] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, and R. Lee. Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 723–737, Berlin, Heidelberg, 2009. Springer.

[34] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd international conference on World Wide Web*, pages 747–758. ACM, 2014.

[35] D. Kontokostas, A. Zaveri, S. Auer, and J. Lehmann. TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. In *Knowledge Engineering and the Semantic Web – 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings*, pages 265–272. Springer, 2013.

[36] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira. An Introduction to the Syntax and Content of Cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background*

Knowledge and Its Applications to Knowledge Representation and Question Answering, pages 44–49. AAAI - Association for the Advancement of Artificial Intelligence, 2006.

[37] M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, and C. Batini. Managing data quality in cooperative information systems. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 486–502. Springer, 2002.

[38] O. Medelyan and C. Legg. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Wikipedia and Artificial Intelligence: An Evolving Synergy, Papers from the 2008 AAAI Workshop*, page 65, 2008.

[39] F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*, volume 2261. Springer Science & Business Media, 2002.

[40] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data Quality Assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[41] E. Sandhaus. Semantic Technology at the New York Times: Lessons Learned and Future Directions. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II*, ISWC'10, pages 355–355, Berlin, Heidelberg, 2010. Springer.

[42] A. Singhal. Introducing the Knowledge Graph: things, not strings. `https://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html`, *retrieved on Aug 29, 2016*, 2012.

[43] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008.

[44] T. P. Tanon, D. Vrandecic, S. Schaffert, T. Steiner, and L. Pintscher. From Freebase to Wikidata: The Great Migration. In *Proceedings of the 25th International Conference on World Wide Web*, WWW 2016, pages 1419–1428, 2016.

[45] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, and B. Aleman-meza. OntoQA: Metric-Based Ontology Quality Analysis. In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.

[46] R. Y. Wang, M. P. Reddy, and H. B. Kon. Toward quality data: An attribute-based approach. *Decision Support Systems*, 13(3):349–372, 1995.

[47] R. Y. Wang and D. M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[48] A. Zaveri, D. Kontokostas, M. A. Sherif, L. Bühmann, M. Morsey, S. Auer, and J. Lehmann. User-driven quality evaluation of dbpedia. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 97–104. ACM, 2013.

[49] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality Assessment for Linked Data: A Survey. *Semantic Web*, 7(1):63–93, 2015.