Semantic Web 1 (2016) 1–5
IOS Press

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

Editor(s): Jie Tang, Tsinghua University, China
Solicited review(s): Zhigang Wang, Beijing Normal University, China; Anonymous; Sebastian Mellor, Newcastle University, U.K.

Michael Färber [*,**], Basil Ell, Carsten Menne, Achim Rettinger [***], and Frederic Bartscherer
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 [21], the term "Semantic Web" already was being used in fields such as Educational Psychology, before it became prominent in Computer Science. Freedman and Reynolds [19], for instance, describe "semantic webbing" as organizing information and relationships in a visual display. Berners-Lee presented his idea of using typed links as vehicle of semantics for the first time at the World Wide Web Fall 1994 Conference under the heading "Semantics," and under the heading "Semantic Web" in 1995 [21].

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 coined by Google in 2012 and is intended for any graph-based knowledge base. We define a **Knowledge Graph** as an RDF graph. An RDF graph consists of a 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 $U \cup B \cup L$. 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 pairwise disjoint. We denote the system that hosts a KG g with h_g.

[*]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.

1570-0844/16/$27.50 © 2016 – IOS Press and the authors. All rights reserved

Further, we define several sets that are used in formalizations throughout the paper:

- C_g denotes the set of classes in g, defined as $C_g := \{x | (x, \texttt{rdfs:subClassOf}, o) \in g \vee (s, \texttt{rdfs:subClassOf}, x) \in g \vee (s, \texttt{rdf:type}, \texttt{owl:Class}) \in g \vee (s, \texttt{wdt:31}, \texttt{wdt:P279}) \in g\}$
- I_g denotes the set of instances in g, defined as $I_g := \{s \mid (s, \texttt{rdf:type}, o) \in g\}$
- P_g^{imp} denotes the set of all implicitly defined properties in g, defined as $P_g^{imp} := \{p \mid (s, p, o) \in g\}$
- R_g denotes the set of all URIs used in g, defined as $R_g := \{x \mid ((x, p, o) \in g \vee (s, x, o) \in g \vee (s, p, x) \in g) \wedge x \in R\}$

Note that knowledge about the knowledge graphs analyzed in the context of 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, in this work we did not derive entailments.

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 data sets on the Web and to interlink these data sets.
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 [14]. 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 comprise 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 were registered at the online dataset catalog `http://datahub.io`[8] and which were tagged as "crossdomain". Besides that, we took further data sets into consideration which fulfilled the above mentioned requirements (i.e., Wikidata). 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 and discuss how the knowledge in these KGs is modeled, stored, and can be 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 aspect when it comes to considering which KG to use in a specific setting.

Furthermore, we provide an evaluation 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.

[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`

[5] See `http://products.wolframalpha.com/api/`

[6] See `https://developers.facebook.com/docs/graph-api`

[7] See `https://www.cia.gov/library/publications/the-world-factbook/`

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

2. We calculate key statistics for the KGs DBpedia, Freebase, Cyc, Wikidata, and YAGO.
3. We analyze DBpedia, Freebase, Cyc, 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 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 3 we describe the selected KGs.
- In Section 4 we analyze the KGs along several key statistics as well as the data quality metrics introduced in Section 2.
- In Section 5 we present our framework for assessing and rating KGs according to the user's setting.
- In Section 6 we present related work on (linked) data quality criteria and on key statistics for KGs.
- In Section 7 we conclude the survey.

2. 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 their 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 Section 2.1.

Data quality (DQ) – in the following interchangeably used with *information quality* – is defined by Juran [29] 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. [40]. They developed a framework for assessing the data quality of data sets in the database context. In this framework, Wang et al. distinguish between *data quality criteria*, *data quality dimensions*, and *data quality categories*.[10] 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. Examples of subjectively measurable data quality criteria are trustworthiness and reputation of a KG in its entirety. Examples of objective data quality criteria are accuracy and completeness (see [39] and also Section 2.1).

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 [40]. For instance, the criteria syntactic validity of the 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* are independent of the use case context; by means of these criteria, one can check whether the information reflects the reality and is logically consistent.
- 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 information 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 [42] worked on data quality in the

[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/`.

[10] The quality dimensions are defined in [40], the subclassification into parameters/indicators in [39, p. 354].

Linked Data context. They make the following adaptations on Wang et al.'s framework:

- Bizer [11] compared the work of Wang et al. [40] with other works in the area of data quality. He thereby complements the framework with the dimensions *consistency*, *verifiability*, and *offensiveness*.
- Zaveri [42] follows Wang et al. [40], but introduces *licensing* and *interlinking* as additional dimensions.

In this article, we use dimensions as defined by Wang et al. [40] and as extended by Bizer [11] and Zaveri [42]. I.e., besides Wang et al.'s dimensions, we incorporate *consistency* and *verifiability* into our framework[11] and extend the category of *accessibility* by the dimension *license* and *interlinking*, as those data quality dimensions get in addition relevant in the Linked Data context.

2.1. Criteria Weighting

When applying our framework to compare KGs, the single metrics and dimensions 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 knowledge graph 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 knowledge graph regarding a quality criterion, 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 $c_1, ..., c_n$:

$$h(g) = \frac{\sum_{i \in [1,n]} w_i \cdot m_i(g)}{\sum_{j \in [1,n]} w_j}$$

[11] *Consistency* is treated by us as a separate dimension. *Verifiability* is treated within the dimension *trustworthiness*, where it is evaluated if provenance vocabulary is used in the KG. The *offensiveness* of KG facts is not considered by us, as it is hard to make an objective evaluation in this regard.

Based on the quality dimensions introduced by Wang et al. [40], 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 6.

2.2. Intrinsic Category

"Intrinsic data quality denotes that data have quality in their own right." [40, p. 6] 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 [40], are subsumed by us into the dimension *trustworthiness*.

2.2.1. Accuracy

Criterion definition. Accuracy is "the extent to which data are correct, reliable, and certified free of error." [40, p. 31]

Discussion. Accuracy is intuitively an important dimension of data quality. Previous work on data quality has mainly analyzed only this aspect [40]. Hence, accuracy has often been used as synonym for data quality [34]. 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 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:

Metric definition. The metric for the dimension *accuracy* is determined by

1. the syntactic validity of RDF documents,
2. the syntactic validity of literals, and
3. the 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 com-

pletely and correctly. Hogan et al. [26] 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.

The RDF can be validated by an RDF validator such as the W3C RDF validator.[12]

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

Syntactic Validity of Literals The aspect of syntactic validity of literals focuses on the adherence of syntactic restrictions w.r.t. literals by means of syntactic rules [20,31]. 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.

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

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

Semantic Validity of Triples The semantic validity of triples is introduced to evaluate whether the meanings of triples with literal values in object position in the KG are semantically correct.

A triple is either semantically correct if it is also available from a trusted source (e.g. Name Authority File) or if it is common sense or if the stated property can be measured or perceived by us firectly.

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

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

2.2.2. Trustworthiness

Definition. Trustworthiness is defined as "the degree to which the information is accepted to be correct, true, real, and credible" [42]. It is used as a collective term for *believability*, *reputation*, *objectivity*, and *verifiability*. These aspects were defined by Wang et al. [40] and Naumann [34, p. 32] as follows:

- **Believability**: Believability is "the extent to which data are accepted or regarded as true, real, and credible." [40, p. 31].
- **Reputation**: Reputation is "the extent to which data are trusted or highly regarded in terms of their source or content" [40, p. 32].
- **Objectivity**: Objectivity is "the extent to which data are unbiased (unprejudiced) and impartial" [40, p. 32].
- **Verifiability**: Verifiability is "the degree and ease with which the data can be checked for correctness" [34, p. 32].

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:

- According to Naumann [34, p. 34], believability is the "expected accuracy" of a data source.
- 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 data set.
- According to Naumann [34], 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 paper would not go so far, since also biased statements could be verifiable.
- Heath et al. [24, p. 52] emphasize that it is essential for trustworthy applications to be able to verify the origin of data.
- In the context of relational databases, Buneman et al. [13] differentiate between *why provenance* and *where provenance*: *Why provenance* answers the question why a particular data item is stored in the database. *Where provenance* answers the question from where the information originated. In this paper, we focus on the *where provenance*, as this aspect is frequently covered in the database and Linked Data area.

Metric. We define the metric for the data quality dimension *trustworthiness* as a combination of trustworthiness metrics on both KG and statement level.

The fulfillment degree of a KG g w.r.t. the dimension *trustworthiness* is measured by the metrics

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

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, (i) the method of data curation and (ii) the method of data insertion (respectively, data source) 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 trusthworthiness on KG level as follows:

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

Note that a user may use different values instead of the 1, 0.75, etc. This also applies for subsequent metrics.

Trustworthiness on statement level The fulfillment of trustworthiness on statement level is determined by an evaluation whether a provenance vocabulary is used. By means of a provenance vocabulary, the source of statements can be stored. Storing source information on statement level is an important precondition to verify statements easily w.r.t. accuracy. The most widely used ontologies for storing provenance information are the Dublin Core Metadata terms[13] with properties such as `dcterms:provenance` and `dcterms:source` and the W3C PROV-O[14] with properties such as `prov:wasDerivedFrom`.

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

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 enable to represent that a person has no children and unknown values enable 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:

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

2.2.3. Consistency

Definition. Consistency implies that "two or more values [in a data set] do not conflict each other" [32].

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, 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

[13] See `http://purl.org/dc/terms/`.
[14] See `http://www.w3.org/ns/prov#`.

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. We also consider the generic "constraints" `rdfs:domain` and `rdfs:range` here. Cardinality constraints limit the number of times a relation may exist per resource. Moreover, via `owl:Functionalproperty` 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 (e.g., an id).

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 (checking the `rdf:type` relation) and 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 are consistent with the class restrictions (e.g., `owl:disjointWith`) specified on the schema level.

We evaluate the criterion as follows. Let CC be the set of all class constraints, defined as $CC := \{(c_1, c_2) \mid (c_1, \text{owl:disjointWith}, c_2) \in g\}$. Furthermore, let $c_g(e)$ be the set of all classes of e in g, defined as $c_g(e) = \{c \mid (e, \text{rdf:type}, c) \in g\}$.

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

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

Consistency of statements w.r.t. relation constraints This metric is intended to measure the degree to which the instance data are consistent with the relation restrictions specified on the schema level (e.g., `rdfs:range`, and `owl:FunctionalProperty`). We evaluate this criterion as follows:

$$m_{conRelat}(g) = \frac{m_{conRelatRange}(g) + m_{conRelatFunct}(g)}{2}$$

Let RC be the set of all relation constraints, defined as $RC := RC_r \cup RC_f$ where $RC_r := \{(p, d) \mid (p, \texttt{rdfs:range}, d) \in g \wedge isDatatype(d)\}$ and $RC_f := \{(p, d) \mid (p, \texttt{rdf:type}, \texttt{owl:functionalProperty}) \in g \wedge (p, \texttt{rdfs:range}, d) \in g \wedge isDatatype(d)\}$.

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

$$m_{conRelatFunctional}(g) = |\{(s, p, o) \in g \mid \exists (p, d) \in RC_f : \neg\exists (s, p, o_2) \in g : o \neq o_2\}| \,/\, |\{(s, p, o) \in g \mid \exists (p, d) \in RC_f\}|$$

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

2.3. Contextual Category

Contextual data quality "highlights the requirement that data quality must be considered within the con-

text of the task at hand" [40, p. 6]. 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 a part of *completeness*.

2.3.1. Relevancy

Definition of dimension. Relevancy is "the extent to which data are applicable and helpful for the task at hand" [40, p. 31].

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*.[15] 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.

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 relevancy of statements among each other statements can be expressed. For instance, given the Wikidata entity "Barack Obama" (`wdt:Q76`) and the relation "position held", "President of the United States of America" has a "PreferredRank" (until this year), while older positions which he has no more are ranked as "NormalRank".

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

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

2.3.2. Completeness

Definition of dimension. Completeness is "the extent to which data are of sufficient breadth, depth, and scope for the task at hand" [40, p. 32].

We include the following aspects of Wang et al. in this dimension:

- *Appropriate amount of data:* Appropriate amount of data is "the extent to which the quantity or volume of available data is appropriate." [40, p. 32]
- *Value-added:* Value-added is "the extent to which data are beneficial and provide advantages from their use" [40, p. 31].

Discussion. Pipino et al. [35] divide *completeness* further 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 are not missing, and
3. *population completeness*, i.e., the extent to which the KG contains all entities.

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 the European stocks. The completeness is, hence, only assessable by means of a concrete use case at hand and with the help of a defined gold standard.

Definition of metric. We stick to the above mentioned distinction of Pipino et al. [35] and define the metric for 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 schema completeness criterion, one can determine the completeness of the schema w.r.t. classes and attributes [35]. The schema is assessed by means of a gold standard. This gold standard would consist of facts which a user deems relevant. In this paper, we exemplify the gold standard with a typical set of cross-domain facts. It comprises (i) basic classes such as persons and locations in different granularities and (ii) basic attributes such as birth date and number of inhabitants. We define the Schema completion $m_{cSchema}$ as the ratio of the number of classes and attributes of the gold standard existing in g no_{clat} and the number of classes and attributes in the gold standard no_{clatg}.

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

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

Column Completeness By means of the column completeness criterion, one can determine the degree by which the attributes of a class, which are defined on the schema level, exist on the instance level of the KG [35]. Assume that $K = \{k_1, ..., k_n\}$ is the set of all considered classes and $R = \{r_1, ..., r_m\}$ the set of considered relations. Then $H = \{(k, r) \in (K \times R) \mid \exists k \in C_g \wedge \exists r \in P_g^{imp}\}$ is the set of all combinations of k and r which can occur on the instance level based on the schema information. We measure the column completeness by evaluating whether that relation r is used on instance level which was defined for a class k on schema level (e.g., the class `dbo:Person` in DBpedia determines the relation `dbo:birthDate`). The column completeness f_{cCol} is defined as the ratio of the number of instances having class k and a value for the relation r no_{kr} and the number of all instances having class k no_r.

$$m_{cCol}(g) = \frac{1}{n} \sum_{(k,r) \in H} \frac{no_{kr}}{no_r}$$

Note that there are also relations which do not need to exist for all entities of the relations dedicated entity type. For instance, not all people need to have a relation `:hasChild` or `:deathDate`.[16] For measuring the column completeness, we selected only those relations for an assessment where a value of the relation typically exists for each entity type the relation is dedicated for.

Population Completeness The *population completeness* metric determines the extent to which the considered KG covers the basic population [35]. The assessment of the completeness of the basic population is performed by a gold standard, which covers both well-known resources (called "short head", e.g., the n largest cities in the world according to the number of inhabitants) and little-known resources (called "long tail"; e.g., municipalities in Germany). We take all resources of our gold standard equally into account:

$$m_{cPop}(g) = \text{Number of resources in gold standard existing in } g \text{ / Number of resources in gold standard}$$

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

2.3.3. Timeliness

Definition of dimension. Timeliness is "the extent to which the age of the data is appropriate for the task at hand" [40, p. 32].

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 [34]. 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 [34]. How timeliness is measured depends on the application context: For some situations years are sufficient, while in other situations one may need days [34].

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 timeliness frequency of a 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. If the RDF export files of the KG are provided with varying timeliness frequencies (i.e., updating intervals), we consider the online version of the KG, since in the context of Linked Data it is sufficient that URIs are dereferenceable.

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

Specification of the validity period of statements Specifying the validity period of statements enables to temporally limit the validity of statements. Regarding 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`[17] and `dcterms:modified`.

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

2.4. Representational Data Quality

Representational data quality "contains aspects related to the format of the data [...] and meaning of data" [40, p. 21]. 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 proposed by Wang et al. [40] in addition are considered by us as being a part of the dimension *interoperability*.

2.4.1. Ease of Understanding

Definition of dimension. The ease of understanding is "the extent to which data are clear without ambiguity and easily comprehended" [40, p. 32].

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 self-explanatory labels, literals in multiple languages,

The dimension *understandability* is determined by the criteria *description of resources*, *labels in multiple languages*, *provisioning of an 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_{Desc}, m_{Lang}, m_{uSer}, and m_{uURI} which are defined as follows.

Description of resources Heath et al. [24,27] suggest describing 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 identify the number of resources for which at least one label or one description is provided, e.g., via `rdfs:label`, `rdfs:comment`, or `schema:description` as no_{desc} and the number of all considered resources no_{cr}.

$$m_{Descr} = \frac{no_{desc}}{no_{cr}}$$

Labels in multiple languages Resources in the KG are described in a human-readable way via labels, e.g. `rdfs:label` or `skos:prefLabel`.[18] The characteristic feature of `skos:prefLabel` is that this kind of label has to 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.

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

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

Self-describing URIs Self-descriptive URIs contribute to a better human-readability of KG data. Sauermann et al.[19] recommend to use short, memorable URIs in the Semantic Web context, which are easier understandable and memorable by humans compared

[17] Using namespace `http://schema.org/`.

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

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

to URIs 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.

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

2.4.2. 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 Wang et al.'s dimensions:

- **Interpretability:** Interpretability is "the extent to which data are in appropriate language and units and the data definitions are clear" [40, p. 32].
- **Representational consistency:** Representational consistency is "the extent to which data are always presented in the same format and are compatible with previous data" [40].
- **Concise representation:** Concise representation is "the extent to which data are compactly represented without being overwhelming" [40, p. 32].

Discussion regarding interpretability. In contrast to the dimension understandability, which focuses on the understandability of KG RDF data towards the user (data consumer), interpretability focuses on the representation from a technical perspective. An example is the evaluation whether the considered KG uses blank nodes. According to Heath et al. [24, p. 17], 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 the best practice to reuse existing vocabulary for the creation of own RDF data. In this way, the potential of reuse can be exploited without any preparation [24, p. 61].

Discussion regarding concise representation.

Heath et al. [24, p. 17] made the observation that the RDF features (i) RDF reification, (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 reification, RDF collections, and RDF container are used.

We can clarify that reification is necessary for stating additional information on statement level. Alternatively, additional relations are necessary (e.g., `dbo:populationAsOf` in DBpedia) which implicitly refer to other statements (e.g., statements of `dbo:populationTotal`).

Definition of metric. The dimension *interoperability* is determined via the 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 *operability* 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 [27][24, p. 17]. 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 is used.

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

Provisioning of several serialization formats The interpretability of RDF data of a KG is increased if besides 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 leads

to represent resources and relations between resources in the Web of Data in a unified way. This increases the interoperability of data [27][24, p. 61] and allows an easy data integration. We measure the criterion of using an external vocabulary by setting the frequency of used external vocabulary in proportion to the number of all triples in the KG:

$$m_{extVoc}(g) = \text{Number of all triples in } g \text{ that use external vocabularies / Number of all triples in g}$$

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 [24, p. 83]. We measure the interlinking on schema level by calculating the ratio to which classes and relations have at least one equivalency link to an external vocabulary (via `owl:sameAs`, `owl:equivalentProperty` or `owl:equivalentClass`) in order to indicate the equivalence to classes and relations that exist external to the KG:

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

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

2.5. 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 those data quality dimensions:

2.5.1. Accessibility

Definition of dimension. Wang et al.'s term *accessibility* result in the additional aspects *availability*, *response time*, and *data request*. Those single aspects are defined as follows:

1. *Accessibility* is "the extent to which data are available or easily and quickly retrievable" [40, p. 32].
2. *Availability* "of a data source is the probability that a feasible query is correctly answered in a given time range" [34].
   According to Naumann [34], 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, so that we need to consider both possibilities.
3. *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, p. 21].
   Note that the response time is dependent on factors such as the query, the size of the indexed data, the data structure, the used triple store, the hardware and so on. Therefore, we do not consider the response time in our evaluations.
4. 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.

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 endpoints*
- *Provisioning of an RDF export*
- *Support of content negotiation*
- *Linking HTML sites with RDF serialization*
- *Provisioning of metadata about a KG*

The fulfillment degree of a KG g w.r.t. the dimension *availability* 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 in RDF should be returned thereby. We assess the dereferencing possibility of resources R by means of taking a sample of URIs: For each of the URIs, the HTTP response status code is analyzed, and it is evaluated whether useful RDF data is returned:

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

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 [34]. 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.[20]

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

Provisioning of public SPARQL endpoints SPARQL endpoints allow the user to perform complex queries (including potentially many entities 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 slide 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, a RDF export data set (RDF dump) can often be downloaded. This data set can then be used to set up a local, private SPARQL endpoint. The criterion here indicates whether an RDF export data set 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. Example is RDF data which is declared as `text/plain` [24, p. 73]. Hogan et al. [26] 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 content type 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 with RDF serialization Heath et al. [24, p. 74] 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 *Autodiscovery pattern* can be included, consisting of the phrase `link rel=alternate`, the indication about the provided RDF content type, and a link to the RDF document.[21] We measure the linking of HTML websites to RDF (resource representation) files by evaluating whether HTML websites contain a

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

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

link as described:

$$m_{HTML\_RDF}(h_g) = \begin{cases} 1 & \texttt{link rel=alternate} \\ & \text{used} \\ 0 & \text{otherwise} \end{cases}$$

Provisioning of metadata about a KG 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[22] [24, p. 48]. For instance, the URI of the SPARQL endpoint can be assigned via `void:sparqlEndpoint` and the RDF export URL can be specified by `void:dataDump`. The metadata can be added as additional facts to the KG or provided as separate VoID file. We measure the provisioning of metadata about a KG by evaluating whether machine-readable metadata about the KG are 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.

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

2.5.2. License

Definition of dimension. Licensing is defined as "the granting of permission for a consumer to re-use a dataset under defined conditions" [42].

Discussion. The publication of licensing information regarding the KGs is important to use the KGs without legal concerns, especially in commercial settings. Creative Commons (CC)[23] published 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 [28]. CC-BY[24] requires specifying the source of the data, CC-BY-SA[25] requires in addition that if the data is published, it is published under the same legal conditions; CC0[26] 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 [28] 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 criteria *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 [27][24, pp. 52].

Licenses can be specified in RDF via relations such as `cc:licence`[27], `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:

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

2.5.3. Interlinking

Definition of dimension. Interlinking is the extent "to which entities that represent the same concept are linked to each other, be it within or between two or more data sources" [42].

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, accord-

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

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

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

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

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

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

ing to Halpin et al. [22], 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 entry of Berlin (in the sense of the capital) links via owl:sameAs relations to three different entries in the KG GeoNames, namely (i) Berlin, the capital,[28], (ii) Berlin, the state,[29], and (iii) Berlin, the city[30]. Moreover, owl:sameAs relations are often created automatically by some mapping function. Mapping errors, hence, lead to a precision less than 100 %.

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

- Interlinking on instance level
- Validity of external URIs

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

Interlinking via owl:sameAs The forth Linked Data principle according to Berners-Lee is the interlinking of data resources so that the user can explore further information. According to Hogan et al. [27], 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 by calculating the ratio to which instances have at least one owl:sameAs link to an external knowledge graph

$$m_{Inst}(g) = \frac{|\{x \in I_g \mid \exists(x, \text{owl:sameAs}, y) \in g\}|}{|I_g|}$$

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 done by owl:sameAs, owl:equivalentProperty, and owl:equivalentClass relations. Web documents are linked by relations such as foaf:homepage and foaf:depiction. Linking to external resources always entails the problem that those links get invalid over time. This can have different causes, such as that the URI is 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). If an RDF resource is linked, we also consider the content type in the header of a corresponding HTTP response, as owl:sameAs statements get invalid if the linked URI is not available anymore and for instance the server makes a forward to the homepage.

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

where $A = \{y \mid \exists(x, \text{owl:sameAs}, y) \in g : (x \in R_g \setminus (C_g \cup P_g) \wedge internal_g(x) \wedge external_g(y))\}$.

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

2.6. Conclusion

We provide 34 criteria classified in 11 dimension which can be applied to assess KGs w.r.t. data quality. Those dimensions themselves are grouped into four categories.

- Intrinsic category
  - Accuracy
    - Syntactic Validity of RDF documents
    - Syntactic Validity of Literals
    - Semantic Validity of Triples
  - 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

[28] http://www.geonames.org/2950159/berlin.html
[29] http://www.geonames.org/2950157/land-berlin.html
[30] http://www.geonames.org/6547383/berlin-stadt.html

    - 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 endpoints
    - Provisioning of an RDF export
    - Support of content negotiation
    - Linking HTML sites with RDF serialization
    - Provisioning of metadata about a KG
  - License
    - Provisioning machine-readable licensing information
  - Interlinking
    - Interlinking via `owl:sameAs`
    - Validity of external URIs

3. Selection of KGs

We consider the following knowledge graphs for our comparative evaluation:

- **DBpedia**: DBpedia[31] is the most popular and 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.[32] DBpedia is created from automatically-extracted structured information contained in the Wikipedia, such as from infobox tables, categorization information, geo-coordinates, and external links. Due to its role as the hub of LOD, DBpedia contains many links to other datasets in the LOD cloud such as Freebase, OpenCyc, UMBEL,[33] GeoNames,[34] Musicbrainz,[35] CIA World Factbook,[36] DBLP,[37] Project Gutenberg,[38] DBtune Jamendo,[39] Eurostat,[40] Uniprot,[41] and Bio2RDF.[42] DBpedia is used extensively in the Semantic Web research community, but is also relevant in commercial settings: companies use it to organize their content, such as the BBC [30] and the New York Times [36]. The version of DBpedia we analyzed is 2015-04.
- **Freebase**: Freebase[43] 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,[44] FMD,[45] and MusicBrainz.[46] Freebase uses a proprietary graph model for storing also complex statements. Freebase shut down its services on June 30, 2015. Wikimedia Deutschland and Google integrate Freebase data into Wikidata via the *Primary Sources Tool*.[47] Further information about the mi-

[31] See `http://dbpedia.org`

[32] There is also DBpedia live which started in 2009 and which is updated when Wikipedia is updated. See `http://live.dbpedia.org/`.

[33] See `http://umbel.org/`

[34] See `http://www.geonames.org/`

[35] See `http://musicbrainz.org/`

[36] See `https://www.cia.gov/library/publications/the-world-factbook/`

[37] See `http://www.dblp.org`

[38] See `https://www.gutenberg.org/`

[39] See `http://dbtune.org/jamendo/`

[40] See `http://eurostat.linked-statistics.org/`

[41] See `http://www.uniprot.org/`

[42] See `http://bio2rdf.org/`

[43] See `http://freebase.com/`

[44] See `http://www.nndb.com`

[45] See `http://www.fashionmodeldirectory.com/`

[46] See `http://musicbrainz.org/`

[47] See `https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool`, requested on Apr 8, 2016.

gration from Freebase to Wikidata are provided in [37]. We analyzed the latest Freebase version as of March 2015.

- **OpenCyc**: The Cyc[48] project started in 1984 as part of 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." While the focus of Cyc in the first decades was on inferencing and reasoning, more recent work puts a focus on human-interaction such as building question answering systems based on Cyc. Since Cyc is proprietary, a smaller version of the KG called OpenCyc[49] was released under the open source Apache license. In July 2006, ResearchCyc[50] was published for the research community, containing more facts than OpenCyc. The version of OpenCyc we analyzed is 2012-05-10.
- **Wikidata**: Wikidata[51] 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 Wikipedia project, including Wikipedia itself.
  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. In the near future, Wikidata will grow due to the integration of Freebase data. The version of Wikidata we analyzed is 2015-10.
- **YAGO**: YAGO – Yet Another Great Ontology – has been developed at the Max Planck Institute for Computer Science in Saarbrücken since 2007. YAGO comprises information extracted from the Wikipedia (e.g., categories, redirects, infoboxes), WordNet[17] (e.g., synsets, hyponymy), and GeoNames.[52] The version of YAGO we analyzed is YAGO3.[53]

[48] See `http://www.cyc.com/`
[49] See `http://www.opencyc.org/`
[50] See `http://research.cyc.com/`
[51] See `http://wikidata.org/`.
[52] See `www.geonames.org/`
[53] See `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/`

4. Comparison of KGs

4.1. Key Statistics

In the following, we give an overview of the 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
- 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 distinct predicates
- Number of instances
- Number of entities per class
- Number of distinct subjects
- Number of distinct objects

In Section 6.2, we provide an overview of related work w.r.t. those key statistics.

4.1.1. Number of Triples and Statements

Number of triples. The number of triples[54] (see Table 1) differs considerably between the KGs: Freebase is the largest KG with over 3.1B triples, while OpenCyc resides the smallest KG with only 2.5M triples. The large size of Freebase can be traced back to the fact that many data sources were integrated into this KG.

Size differences between DBpedia and YAGO. As both DBpedia and YAGO were created by extracting semantically-structured information from Wikipedia, the significant difference between their sizes (in terms of triples) is noteworthy. YAGO integrates the statements from different language versions of Wikipedia in one single KG while for the canonical DBpedia data set (which is used in our evaluations) solely the English Wikipedia is used. Besides that, YAGO contains contextual information and detailed provenance information. Contextual information are for instance the anchor texts of all links within Wikipedia where the respective entity is linked. For that, the relation `hasWikipediaAnchorText` (330M triples in total) is used. The provenance information of single statements is stored in a reified form. In particular, the relations `extractionSource` (161.2M triples) and

[54] Measured via the SPARQL query `select count(*) where ?s ?p ?o.`

`extractionTechnique` (176.2M triples) are used for that.

Influence of reification on the number of triples. As we will see in Section 4.2.8 in more detail, YAGO and Wikidata use reification for data modeling. In case of YAGO, however, the number of triples is due to that only increased to a very limited extend, since YAGO data is provided in N-Quads.[55] The additional column (in comparison to triples), the IDs, by which the triple become uniquely identified, are normally considered as comment and therefore are not imported into the triple store. For Wikidata, things are different: The Wikidata RDF export is stored in N-Triples format, so that reification has a great impact on the number of instantiations of statements, and therefore, on the number of triples in total. In our evaluation, we counted 74,272,190 instances of the class `wdo:Statement`.[56] The instantiations make up about a tenth of all triples.

4.1.2. Classes and Domains

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 (plus the most general class).[57] Since Freebase does not provide any class hierarchy with `rdfs:subClassOf` relations and since Wikidata does not mark 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 1).

Number of classes in YAGO and DBpedia How does it come to this gap between DBpedia and YAGO w.r.t. the number of classes? The YAGO category taxonomy, which serves as set of classes, is automatically created by the Wikipedia categories and the set of WordNet synsets. The DBpedia ontology, in contrast, is very small, since it is created manually, based on the mostly used infobox templates in Wikipedia. In total, the DBpedia KG contains 444,895 classes which are connected via `rdfs:subClassOf` to a taxonomy. Those additional classes originate from the imported YAGO categories and therefore use the namespace `http://dbpedia.org/class/yago/`.

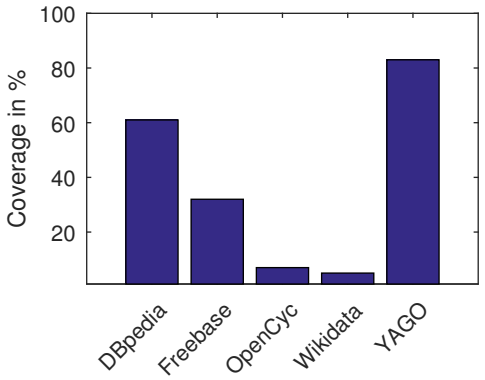


Fig. 1. Coverage of classes with at least one instance per class; per KG.

Ratio of classes which are used in the KG Figure 1 shows the coverage of classes with at least one instance per KG. Interestingly, none of the KGs achieves a very high coverage. YAGO performs best with 82.6% coverage, although it contains the highest number of classes. This can be traced back to the fact that YAGO classes were created by heuristics which select Wikipedia categories with multiple instances. OpenCyc (with 6.5%) and Wikidata (5.4%) come last in the ranking. Especially Wikidata contains relatively many classes (second most in our ranking above), out of which only few of them are used at all.

Figure 2 shows the distribution of classes regarding the number of the corresponding instances. Note the logarithmic scale on the axis of ordinates. We can recognize an almost linear curve progression for all KGs except for DBpedia. For DBpedia, the line drops at around class 250 exponentially. This means that only a small part of the already small DBpedia ontology is used in reality (i.e., on the instance level).

Domains Tartir [38] proposed to measure the covered domains by determining the class importance, i.e., the proportion of instances belonging to one or more subclasses of the respective domain in comparison 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*, *or-*

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

[56] Using the namespace `http://wikidata.org/ontology#`.

[57] 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.

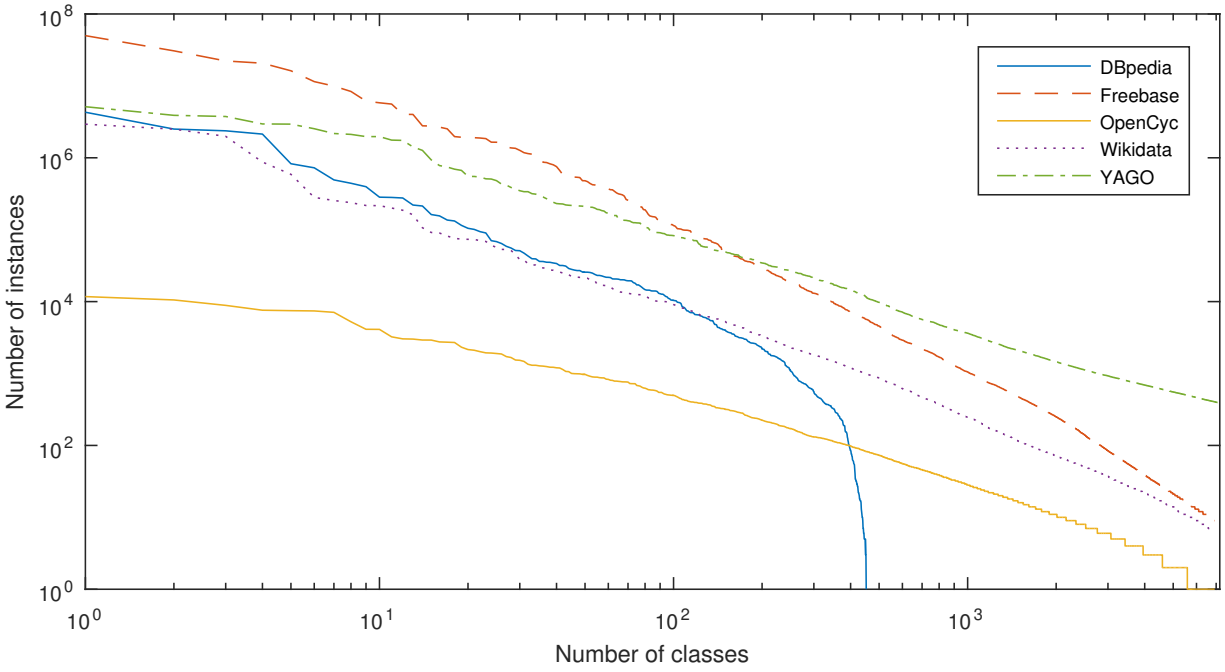


Fig. 2. Distribution of classes w.r.t. the number of their corresponding instances; shown per KG.

ganizations, *geography*, and *biology* (see our website for examples of classes per domain and per KG). This list of domains was created by aggregating the most frequent domains in Freebase.

The manual domain assignment is necessary to obtain a consistent classification of the classes into the domains across all considered KGs. In Freebase, instances can appear in various domains. For instance, an intersection of the classes `/music/artist` and `/people/person` is obvious.

We retrieve the number of unique instances per domain and per KG via SPARQL queries, given the most frequently used classes per KG and domain at hand.

Figure 4 shows the number of unique entities per domain in the different KGs. As the reader can see in Figure 3, we obtained a coverage of about 80% for all KGs except Wikidata. Via this coverage scores we measure the performance of the method for selecting the classes, i.e. the reach of our evaluation. It is calculated as the ratio of the number of unique entities of all considered domains of a KG divided by the number of all entities of this KG.[58] If the ratio is at 1.0, we could assign all instances of a KG in one of the domains.

[58] 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. 3. Coverage of entities: Number of unique entities of all five domains divided by the number of all entities in the KG.

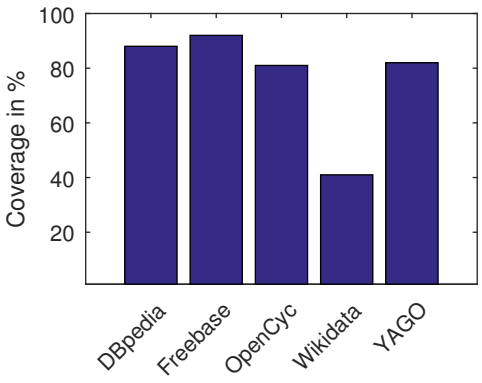


Figure 5 shows the relative coverage of each domain in each KG, i.e., the proportion of the number of instances of each domain of each KG to the total number of instances of the KG. A value of 100% means that all instances reside in one single domain. In case of Freebase, 77% of all instances are in the media domain. The statements of those instances notably originate from data imports such as from MusicBrainz.org. In DBpedia and YAGO, the domain of people is the largest domain (50% and 34%, respectively). Peculiar is also the higher coverage of YAGO regarding the geography domain compared to DBpedia. As one reason

Fig. 4. Number of entities per domain.

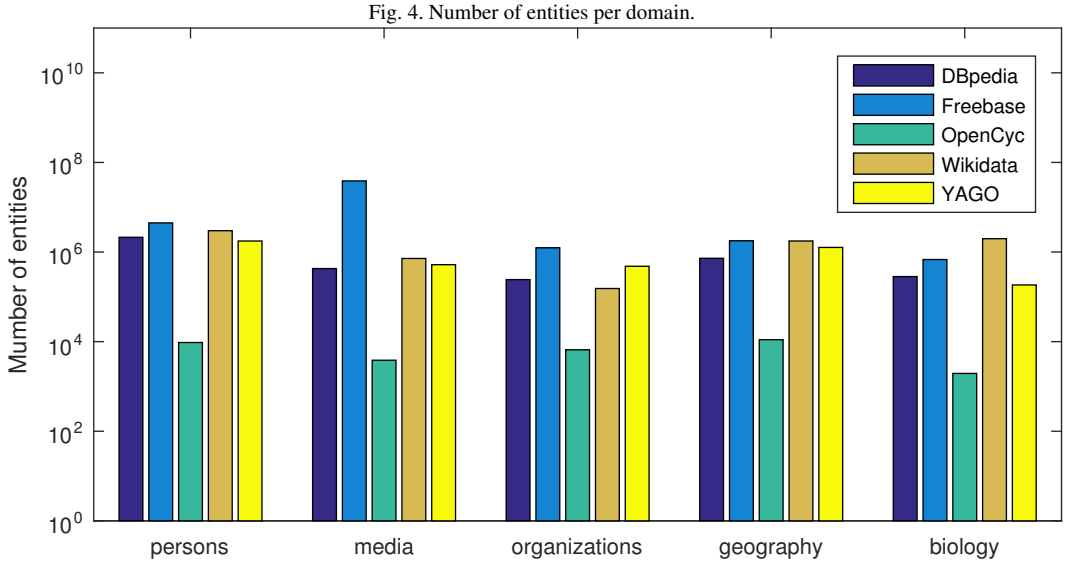
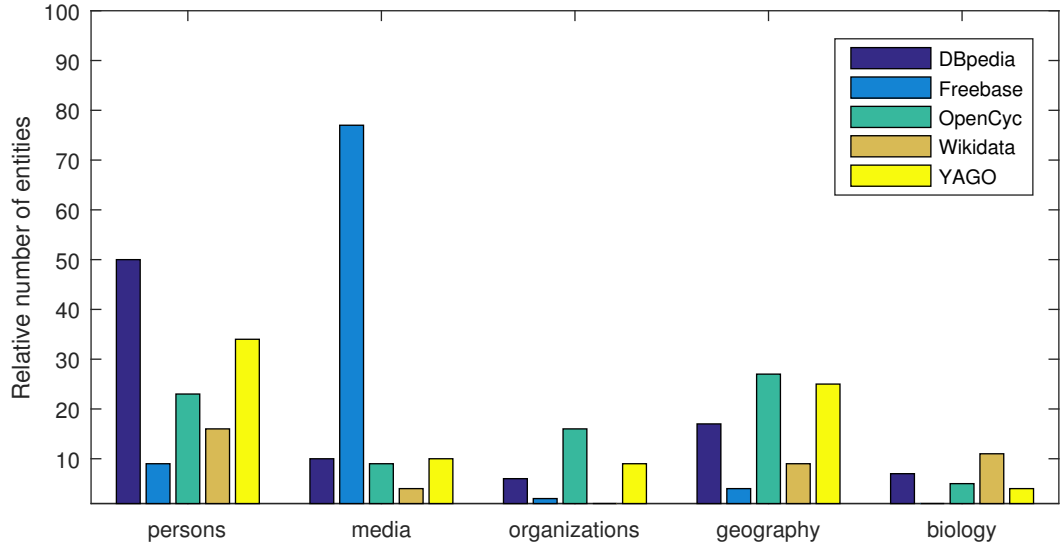


Fig. 5. Relative number of entities per domain.



for that we can point out the data import of GeoNames into YAGO.

4.1.3. Relations and Predicates

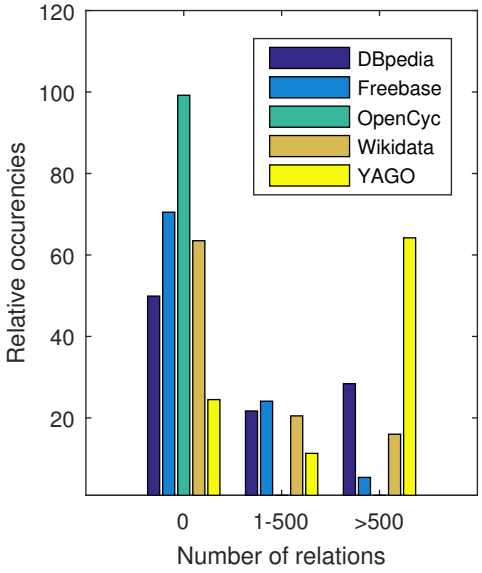
In this work, we differentiate between *relations* and *predicates*: Relations (interchangeably used with "properties") is (proprietary) vocabulary defined on the schema level (i.e., T-Box). In contrast, we use predicates to denote the connections on the instance level (i.e., A-Box), such as `rdf:type`, `owl:sameAs`, etc.

It is important to distinguish the key statistics for relations and for predicates, since otherwise those relations remain unconsidered during the identification of the unique predicates which are not used in the KG, i.e., not used on the instance level.

Identification of Relations and Predicates

We identify the relations of each KG by means of classifying proprietary relations in the KG namespace via `rdf:Property`, `rdfs:Property`, and via OWL classes such as `owl:FunctionalProperty` and in the special case of Wikidata via the class `wdo:Property`. The set of unique predicates per KG is determined by using the terms on the predicate position of N-Triples (and making them unique).

Fig. 6. 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.



Key statistics regarding relations and predicates for identifying all relations of the KGs on the schema level.

Ranking regarding predicates Regarding the predicates, Freebase exhibits by far the highest number of unique predicates (784,977) among the KGs (see Table 1). OpenCyc shows only 165 unique predicates, which is the lowest value in this comparison.

Freebase Freebase exhibits a high number of relations. This can be traced back to the integration of several data bases. About a third of all relations (23,728 out of 70,902 relations) in Freebase are declared as inverse of other relations (via `owl:inverseOf`).[59] However, as visible in Figure 6, only 3,825 (5%) relations are used more than 500 times and about 70 % are not used at all. Freebase also shows a large number of different predicates, but 95% (743,377 predicates) are used only once.

Wikidata Wikidata provides a relatively small set of relations and predicates. 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 predicates as it was possible in Freebase. Instead, predicates first need to be proposed[60] and then get accepted by the community only if certain criteria are met. One of those criteria is that the new predicate is used presumably at least 100 times. Note further that the Wikidata RDF export contains the supplementaries v, s, q, and r for reification of statements. "s" indicates that the term in the object position is a statement, "v" refers to a value, "r" refers to a reference, and "q" to a qualifier. Relations are always used together with these supplements.

DBpedia The set of DBpedia relations is also quite limited. The DBpedia KG contains, however, 58,776 instances of the class `rdf:Property` and within the namespace `dbp`.[61] Those are the non-mapping based properties, i.e., properties originating from the unfiltered, generic InfoboxExtractor. The names of the mapping-based properties and of the non-mapping based properties are not aligned[62] and sometimes also overlap.[63] The difference between the 59K relations instantiated via `rdf:Property` and the 60K different predicates results from using external vocabulary such as `owl:sameAs`.

YAGO For YAGO, we measure 106 distinct relations in the namespace `yago:` and 88,736 distinct predicates. Similar to DBpedia, the large discrepancy can be explained by the fact that the infobox relations are not declared as relations on the schema level.

Relations of DBpedia vs. YAGO Although relations are curated manually for YAGO and DBpedia, the size of the set of relations differs significantly. Hoffart et al. [25] mention the following reasons for that:

1. Relations in the DBpdia ontology are partially very special and fine-grained. For instance, there exists the relation `dbo:aircraftFighter` between `dbo:MilitaryUnit` and `dbo:MeanOfTransportation`. YAGO, in contrast, uses only the generic relation `yago:hasCreated`. We can confirm this observation in that way that about 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 (see Figure 6).

[59] An example are the inverse relations "music/artist/album" and "/music/album/artist."

[60] See `https://www.wikidata.org/wiki/Wikidata:Property_proposal`.

[61] `http://dbpedia.org/property/`.

[62] E.g., 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`.

[63] E.g., `dbp:alias` ans `dbo:alias`.

2. The DBpedia ontology allows several relations for birth dates (e.g., `dbo: birthDate` and `dbo:dbobirthYear`), while in YAGO only the relation `yago:birthOnDate` is used. Incomplete date specifications (e.g., only the year is known) are specified by wildcards ("#").
3. YAGO has no relations explicitly specified as inverse such as between `dbo:parent` and `dbo:child`.
4. YAGO uses the SPOTL(X) format for extending the triple format for specifications for time and location. Hence, no dedicated relations such as `dbo:distanceToLondon` or `dbo:populationAsOf` are necessary.

Predicates of DBpedia vs. YAGO YAGO differs from DBpedia also significantly in the number of predicates. YAGO contains more predicates, since infobox attributes from the various language versions of Wikipedia are aggregated into one KG.[64] DBpedia, in contrast, provides localized KGs.

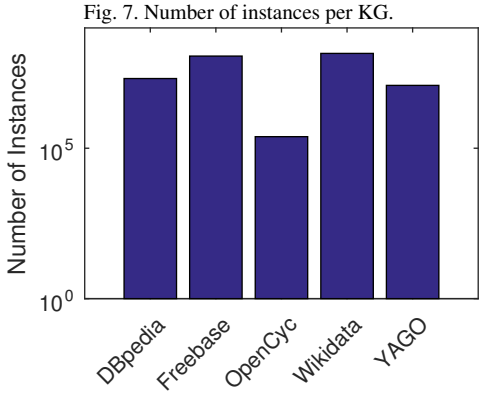
OpenCyc Compared to the 18,028 defined relations for OpenCyc, we measured only 164 distinct predicates. 99.2% of the relations were never used. We assume that those relations are used just within Cyc.

4.1.4. Instances and Entities

We distinguish between entities and instances: Instances are resources having an entity type (i.e., having a `rdf:type` relation). Entities are those instances which are in addition represented as real world objects. Instances of `wdo:Statement` etc. are therefore not regarded as entities.

Evaluation method. We identify the different instances by retrieving the subjects of all triples where `rdf:type` is the predicate. We identify the different entities from the set of instances which in addition belong to `owl:Thing` in DBpedia and YAGO, to `freebase:common.topic` in Freebase, and to `wdo:Item` in Wikidata. In OpenCyc, `cyc:Individual` corresponds to `owl:Thing`, but not all entities are classified in this way. We estimate the set of entities in OpenCyc by considering all classes which comprise more than 300 instances and which contain entities.[65] In the same way we selected the classes manually for the analysis in Section 4.1.2; we obtained 41,029 unique instances out of 242,383, which corresponds to a coverage of 17%.

Fig. 7. Number of instances per KG.



Ranking w.r.t. the number of instances Wikidata comprises the highest number of instances per class, OpenCyc the fewest (see Figure 7).

Ranking w.r.t. the number of entities Table 1 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 41,029 identified entities.

Wikidata exposes relatively many instances in comparison to the entities, since it uses reification and since it stores the Wikipedia sitelinks. Via instantiation of statements via `wdo:Statement` over 74M instances are created.

Freebase exposes also relatively many instances in comparison to the entities, since each node is represented by a M-ID and by an unchangeable GUID. In the RDF export of Freebase, about 35.8 M GUID nodes are instantiated as `xsd:integer`.

Differences of entities The reason why the KGs show quite different numbers of entities lies in the sources of the KG knowledge. We illustrate this with the music domain as example:

1. *Freebase* was created mainly from data imports (e.g., imports from MusicBrainz.com). Therefore, the domain of media and especially song release tracks are covered very well (77% of all entities are in the media domain (see Section 4.1.2), out of which 42% are release tracks (`/music/release_track`)).

[64] The language of each attribute is encoded in the URI, e.g., `infobox/de/fläche` and `infobox/en/areakm`.

[65] Examples for those classes are "individual" (`cyc:Mx4rvVjaApwpEbGdrcN5Y29ycA`), "movie" (`cyc: Mx4rv973YpwpEbGdrcN5Y29ycA`) und "city" (`cyc:Mx4rvVjnZ5wpEbGdrcN5Y29ycA`) (namespace cyc: urlhttp://sw.opencyc.org/concept/. This selection method neglects abstract classes such as "type of objects" (`cyc:Mx4rvWXYgJwpEbGdrcN5Y29ycA`.

Therefore, 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 the 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 (e.g., 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 language-specific DBpedia version), this information is often available (e.g., 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 of Helene Fischer are not in the YAGO KG, although the song "Lass' mich in dein Leben" exists in the German Wikipedia since May 2014.[66]
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 such as "The Lady in My Life". Note, however, that Wikidata does not provide all artist's works such as from Helene Fischer.
5. *OpenCyc* has only few entities in comparison to the other KGs, as it mainly consists of a taxonomy.

[66] YAGO3 is based on the Wikipedia dump of 2014-06-26, see http://www.mpi-inf.mpg.de/de/departments/databases-and-information-systems/research/yago-naga/yago/archive/.

Fig. 8. Average number of entities per class per KG.

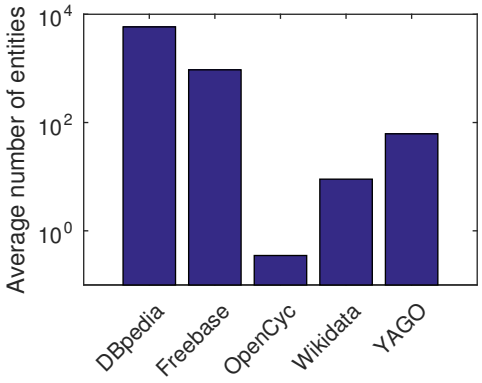
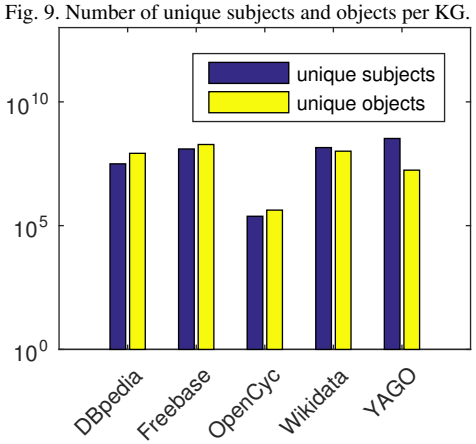


Fig. 9. Number of unique subjects and objects per KG.



Average number of entities per class Figure 8 shows the average number of entities per class. 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).

4.1.5. Subjects and Objects

Evaluation method We measure the number of unique subjects by counting the unique resources on the subject position of N-Triples, excluding blank nodes. Analogously, we measure the number of unique objects by counting the unique resources on the object position of N-Triples, excluding literals. We also measure the number of blank nodes and the number of literals separately.

Ranking of KGs w.r.t. unique subjects Figure 9 shows the number of unique subjects per KG. YAGO contains the highest number of different subjects,

while OpenCyc contains the fewest. Blank nodes are used in Wikidata and OpenCyc.

Peculiarities Surprising is the high number of unique subjects in YAGO compared to the number of instances (see Figure 9 and Table 1). This is caused by the representation form of provenance information: For that, ids are used on the subject position of N-Triples and lead to 308M unique (additional) subjects.[67] In the RDF export of YAGO, those IDs are commented out in order to ensure compatibility to Turtle/N-Triples.

Wikidata provides a ratio of unique subjects to instances of about 1. A reason for that might be that each subject resource gets instantiated, e.g., as entity (`wdo:Item`), statement (`wdo:Statement`), or sitelink (`wdo:Article`). Especially the 45M interwiki links (linking to Wikipedia, Wikiquote, etc.) contribute to the high ratio, since on the subject position the target URI of the Wikimedia project is written, instead of the object position, as it is common for links like `owl:sameAs`.[68]

Ranking of KGs regarding number of objects Figure 9 displays also the number of unique objects per KG. Freebase shows the highest score in this regard, OpenCyc again the lowest.

Noteworthy is especially the difference between DBpedia and YAGO. One reason for the higher value regarding DBpedia might be the considerably higher number of external links via `owl:sameAs` in case of DBpedia (29.0M vs. 3.8M links).

4.1.6. Summary of Key Statistics

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

- *Number of triples and statements*: Freebase is the largest KG in terms of number of triples, OpenCyc is the smallest. As Wikidata RDF export is stored in N-Triples format, reification has a great impact on the number of instantiations of statements, and, hence, of triples (about a tenth of all triples).
- *Number of classes*: None of the KGs achieves a high coverage value regarding the classes (classes with at least one instance in the KG). YAGO obtains the best coverage, despite its high number of classes. This can be traced back to the heuristics used for selecting the classes. Wikidata contains relatively many classes, but from them only few are used at all. Also regarding DBpedia, only a small part of the already small, manually created DBpedia ontology is actually instantiated.
- *Coverage of domains*: In DBpedia and YAGO, the domain people is the largest (50% and 34%, respectively). YAGO shows a high coverage regarding the geography domain due to imports such as GeoNames. 77% of all instances in Freebase are in the media domain due to imports such as MusicBrainz.
- *Relations and Predicates*: Many relations and predicates are rarely used in the respective KGs: Only 5% of the Freebase relations are used more than 500 times and about 70% are not used at all. 95% of the predicates are only used once. 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 within Cyc only.
- *Instances and Entities*: Freebase contains by far the highest number of entities and also covers entities well which are primarily known only in non-English speaking countries. Wikidata exposes relatively many instances in comparison to the entities, since it uses reification and since it stores the Wikipedia sitelinks. YAGO provides a high number of unique subjects, as it contains provenance information. DBpedia has a high number of unique objects, likely due to the high number of external links such as `owl:sameAs`.

4.2. Data Quality Analysis

We now present the results obtained by applying the DQ metrics (as introduced in Section 2.1) for the KGs.

4.2.1. Accuracy

Syntactic validity of RDF documents

Evaluation method. For evaluating the syntactic validity of RDF documents, we dereference the resource "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 val-

[67] For the provenance information, the relations `yago:extractionSource` and `yago:extractionTechnique` are used, leading to 308,619,422 unique subjects such as yago:id_6jg5ow_115_lm6jdp.

[68] An example triple is `https://en.wikipedia.org/wiki/Hapiness schema:about wdt:Q8`.

Table 1
Summary of key statistics

	DBpedia	Freebase	Opencyc	Wikidata	YAGO
Number of triples	411 885 960	3 124 791 156	2 412 520	748 530 833	1 001 461 792
Number of classes	736	53 092	116 822	302 280	569 751
Number of relations	58 776	70 902	18 028	1874	106
Unique predicates	60 231	784 977	165	4839	88 736
Number of entities	4 298 433	49 947 799	41 029	18 697 897	5 130 031
Number of instances	20 764 283	115 880 761	242 383	142 213 806	12 291 250
Avg. number of entities per class	5840.3	940.8	0.35	61.9	9
Unique subjects without blank nodes	31 391 413	125 144 313	239 264	142 213 806	331 806 927
Number of blank nodes	0	0	21 833	64 348	0
Unique non-literals in object position	83 284 634	189 466 866	423 432	101 745 685	17 438 196
Unique literals in object position	161 398 382	1 782 723 759	1 081 818	308 144 682	682 313 508

Table 2
Syntactic validity of RDF documents

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

idator.[69] 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.[70]

Result. All considered KGs provide the syntactic validity of RDF documents. However, 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, in contrast, contained in the newer standard ISO 639-3, so that they are actually valid.

Syntactic validity of literals **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 they 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.

The KG OpenCyc is not taken into account for this criterion. Although OpenCyc comprises 1,081,818 literals in total, these literals are essentially labels and descriptions (given via `rdfs:label` and `rdfs:comment`) and are not bounded to special formats. Hence, OpenCyc has no syntactic invalid literals and is assigned the metric value 1.

As far 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. Thereby, standard data types such as `xsd:date` can be validated easily. The flexible validation method allows also the validation of literals with different data types for a relation. In DBpedia, for instance, data for the relation `dbo:birthdate` is stored both as `xsd:gYear` and as `xsd:date`. If the literal has no type or if it is of type `xsd:String`, the literal is evaluated by a regular expression, which was created manually (see below, depending on the relation considered). For each of the three relations selected for the evaluation, we created a sample of 1M literal values per KG, as long as the respective KG contains so many literals.

Evaluation results.

Date of Birth For Wikidata, DBpedia, and Freebase, all verified literal values (1M per KG) were syntactically correct. 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.

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

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

For YAGO, we detected 519,049 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 correct syntax of `xsd:date`. Obviously, the syntactic invalidity of literals is accepted by the YAGO KG publishers in order to keep the number of relations low.[71]

Number of inhabitants For DBpedia, YAGO, and Wikidata, we evaluated the syntactic validity of the number of inhabitants by means of the data types `xsd:nonNegativeInteger`, `xsd:decimal`, and `xsd:integer` of 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, period, and comma. The values for the number of inhabitants were valid in all KGs.

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[72] released a regular expression for validating ISBN in its different forms, which we use in our experiments.

In Freebase, 698,736 ISBN numbers are available. Out of them, 38 were assessed as syntactically incorrect. Typical mistakes were too long numbers[73] and wrong prefixes.[74] In case of Wikidata, 18 of the 11,388 ISBN numbers were syntactically invalid. However, some invalid numbers have already been corrected since the time of evaluation. This indicates that the Wikidata community does not only care about inserting new data, but also increasing the quality of the 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 24,184 literals. 7,419 of them were assessed as syntactically incorrect. In many cases of the incorrect literals, comments next to the ISBN numbers in the info-boxes of Wikipedia led to inaccurate extraction of data, so that the comments are either extracted as additional ISBN number facts[75] or together with the actual ISBN numbers as coherent strings.[76]

Semantic Validity of Triples **Evaluation method.**

Evaluating the semantic validity is hard, even if a random sample set is evaluated manually, e.g., via crowd-sourcing [2]. Kontokostas et al. [31] propose a test-driven evaluation of Linked Data quality where test cases are automatically instantiated based on schema constraints or semi-automatically enriched schemata. The authors propose a test-case generator for measuring, among other things, the ratio of valid domains and ranges of relations. For instance, an interval specifies the valid height of a person and all triples which lie outside of this interval are evaluated manually. Beyond that, literals such as the ISBN number can be evaluated w.r.t. their semantic validity automatically by means of their check sum. This method requires the considered literals to be consistently formatted.

In this work, we use a semi-automatic method: First, rules for evaluating the semantic validity of the person names, the birth dates, and the number of inhabitants are created. Literals which do not match with the rules are then evaluated manually.

Person names

Evaluation method. We evaluated the semantic validity of person names[77] by means of a sample of 100,000 literals. The literals were selected in DBpedia based on `foaf:name`, in YAGO based on `yago:hasFamilyName` for the family name and based on `yago:hasGivenName` for the given name. For Freebase, Wikidata and OpenCyc, `rdfs:label` was used in combination with the constraint that the resource is instance of a person class.[78] Literals with more than two characters were assessed as valid, literals with two or fewer characters were evaluated manually.

Evaluation result. All considered KGs scored well in this evaluation. Out of the 100,000 evaluated person names from DBpedia, 73 literals had two or fewer characters. However, 64 of them consist of Chinese

[71] 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:xsd:gYearMonth*, and using them only if the whole date is not known.

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

[73] E.g., the 16 digit ISBN 9789780307986931 (`m/0pkny27`).

[74] E.g., prefix 294 instead of 978 regarding 2940045143431.

[75] An example is `dbr:Prince_Caspian`.

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

[77] We used person names instead of ISBN numbers, since we wanted to check the semantic validity via a checksum automatically. However, this requires that the literal values to be checked have a common format. This is not given for ISBN numbers in the considered KGs as analyses have shown.

[78] I.e., of type `freebase:people.person` in case of Freebase and of type "human" (wdt:Q5) in case of Wikidata.

characters and one literal is an alias.[79] In total, 8 invalid literals were identified.[80] The corresponding errors can be traced back to errors in the DBpedia extraction framework. For YAGO, we found 158 literals with two or fewer characters. Our manual investigation revealed, however, that all of them are correct names. The Wikidata sample contained 110 short literals. They were non-arabic letters and the names are presumably correct. For Freebase, 241 literals were marked as invalid. 214 of them were written in non-latin letters. The remaining literals are often stage names such as "MB" (`m/0x90hz6`).

Dates of birth

For evaluating the dates of birth we reused the dates of birth sample from the syntactic validity evaluation. Here, we achieved similar results than for the evaluation of the person names. Interesting is that we found 36 wrong dates in YAGO and 30 wrong dates in Wikidata, such as February 31.

Number of inhabitants

Getting to know the actual number of inhabitants and thereby semantically validating the number of inhabitants is very hard, since different sources may state different values. We therefore used a valid domain range for assessing the number of inhabitants. In our evaluation, all literal values passed the validity check here.[81]

Conclusions

Since we measured low numbers of semantic invalid literals in the relatively large sample sets (100,000 values for each KG and attribute), the metric function returns 1 for all KGs (Table 2). This corresponds to the highest fulfillment score. During our evaluation, we identified several noteworthy errors (which could be used for improving the extraction systems; see the detailed analysis in our wiki). However, a qualitatively better evaluation is only achievable by a manual evaluation – as it was performed for YAGO2 by the YAGO developer team. In this evaluation, assessing 4,412 statements manually resulted in an accuracy of 98.1% (with weighted averaging: 95%).[82] The measured accuracy values were generalized per relation and are stored in the KG as additional facts via the relation `yago:hasConfidence`.

Table 3
Measured values for 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

4.2.2. Trustworthiness

The values of the metrics are shown in Table 3.

Trustworthiness on KG level Regarding the trustworthiness of a KG in general, we can differentiate between the method of how new data is inserted into the KG and the method of how existing data is curated. The KGs differ considerably w.r.t. these dimensions. Cyc is edited (expanded and modified) exclusively by a dedicated expert group. The free version, OpenCyc, is derived from Cyc and only the data of a local mirror 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.[83] However, new data is not just inserted, but needs to be 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 does not need community approvals. The knowledge of both DBpedia and YAGO is extracted from Wikipedia, but DBpedia differs from YAGO w.r.t. the community involvement: Any user can engage in the mappings of the Wikipedia infobox templates to the DBpedia ontology in the mapping wiki of DBpedia[84] and in the development of the DBpedia extraction framework.

In total, OpenCyc obtains the highest value here, followed by Wikidata.

[79] "Mo" was used for `dbr:Maurice_Smith_(kickboxer)`.

[80] Among them, "or" for `dbr:Yunreng`, "*" for `dbr:Carmine_Falcone`, and "()" for `dbr:Blagovest_Sendov`.

[81] In the sample, there were also cities with no inhabitants. However, those cities (such as "Miklarji" (m/0bbx9zd)) indeed do not have inhabitants.

[82] See `http://www.mpi-inf.mpg.de/de/departments/databases-and-information-systems/research/yago-naga/yago/statistics/`, requested on Mar 3, 2016.

[83] 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`).

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

Trustworthiness on statement level We defined the metric for trusthworthiness on statement level via evaluating whether provenance information is stored for statements in the KG. The picture is mixed: DBpedia uses the relation `prov:wasDerivedFrom` from the W3C-PROV-O ontology to represent the sources of the entities and their statements. However, as the source are Wikipedia articles in all cases (e.g., `http://en.wikipedia.org/wiki/Hamburg` for `dbr:Hamburg`) and since all DBpedia entities have a corresponding Wikipedia page, this provenance information are trivial and the fulfillment degree is, hence, of rather formal nature. Furthermore, in the DBpedia ontology, `prov:wasDerivedFrom` and `dcterms:source` triples are used to connect the ontology with the DBpedia mapping wiki.[85] 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 extraction technique (via `yago:extractionTechnique`; e.g., "Infobox Extractor" or "CategoryMapper"). The number of indications 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 statements.

In Wikidata several relations can be used for referring to sources, such as `wdt:P143` ("imported from"), `wdt:P248` ("stated in"), and `wdt:P854` ("reference URL").[86] Note, that "imported from" relations are used for automatic imports but that statements with such a reference are not accepted ("data is not sourced").[87] To source data, the other relations, "stated in" and "reference URL", can be used. The number of all stored references in Wikidata[88] is 971,915. Based on the number of all statements,[89] 74,272,190, this corresponds to a coverage of 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 de facto are missing, a manual evaluation would be necessary. However, such an evaluation would be presumably highly subjective.

[85] See `http://mappings.dbpedia.org`, requested Mar 3, 2016.

[86] All source relations are instances of `Q18608359`.

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

[88] `?s a wdo:Reference`

[89] `?s a wdo:Statement`

Table 4
Check of consistencies.

	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	0.99

Freebase uses proprietary vocabulary for representing provenance: via Compound Value Types (CVT), relations of higher degree can be expressed [37]. For the relation `/location/statistical_region.population`, for instance, a concrete value and a corresponding source (`/measurement_unit/dated_integer/source`) can be stored via an intermediate node.

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

Indicating unknown and empty values This criterion highlights the subtle data model of Wikidata and Freebase in comparison to the data models of the other KGs: Wikidata and Freebase allow 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`.

In YAGO, non-exact dates are representable via wildcards (e.g., "1940-##-##", if only the year is known). Note, however, the invalidity of such strings as date literals (see Section 4.2.1). Unknown dates are not supported by YAGO.

4.2.3. Consistency

Check of schema restrictions during insertion of new statements The values of the metric $m_{checkRestr}$ regarding the restrictions during the insertion of new statements are shown in Table 4. 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, for the relation "date of birth" (`wdt:P569`) Wikidata expects an input of a date in syntactically valid form. DBpedia, OpenCyc and YAGO have no schema restriction during insertion.

Consistency of statements w.r.t. class constraints

Evaluation method For evaluating the consistency of class constraints we used the relation `owl:dis-`

jointWith, since this is the only relation which is used by more than half of the considered KGs. We only considered direct instantiations here: If there is, for instance, the triple (dbo:Agent, owl:disjointWith, dbo:Place), then there should be no instances ?s with ?s rdf:type dbo:Agent and simultaneously ?s rdf:type dbo:Place.

Evaluation results The scores are shown in Table 4. Freebase and Wikidata do not specify any constraints with owl:disjointWith. Hence, those two KGs have no inconsistencies w.r.t class restrictions so that we assign the metric value 1 here. 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 1200 instances exist which are both a dbo:Agent and a dbo:Place. YAGO contains 42 constraints, dedicated mainly for WordNet classes.

Consistency of statements w.r.t. relation constraints
Evaluation method Here we used the relations rdfs:range and owl:FunctionalProperty, as they are used in more than every second considered KG. We only consider datatype properties since consistencies regarding object properties would require to distinguish Open World assumption and Closed World assumption.

In the following, we consider the fulfillment degree for the two mentioned relations separately:

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 page of the relations on https://wikidata.org, users can indicate the allowed entity types via "One of" statements. However, this is not part of the Wikidata data model.

DBpedia obtains the highest measured value w.r.t. rdfs:range. Most inconsistencies here are due to inconsistent data types. For instance, the relation dbo:birthDate requires a data type xsd:date. In about 20.3% of those relations the data type xsd:gYear is used, though.

Also, YAGO, Freebase, and OpenCyc contain mainly inconsistencies since they use data types on the schema level which are not used consistently on the instance level. For instance, YAGO specifies the data types yago:yagoURL and yago:yagoISBN in its schema. On the instance level, however, either no data type is used or the unspecific data type xsd:string.

Table 5
Create a ranking of statements

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

FunctionalProperty If a relation is instantiated as owl:FunctionalProperty, this relation should only be used at most once per resource.

The restriction via owl:FunctionalProperty is used by all KGs except Wikidata. On the talk pages about the relations on the Wikidata online platform, users can specify this kind of 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.

4.2.4. Relevancy

Creating a ranking of statements Only Wikidata supports the creation of a ranking of statements (see Table 5): Each statement is ranked and can either reach the value “preferred rank” (wdo:PreferredRank), “normal” (wdo:NormalRank), or “deprecated” (wdo:PreferredRank). The preferred rank corresponds to the up-to-date value or the consensus of the Wikidata community w.r.t. this statement. Freebase does not provide any ranking of statements, entities, or relations. However, the meanwhile shutdown Freebase Search API provided a ranking for resources.[90]

4.2.5. Completeness

We evaluated the completeness by a created gold standard, which is available online.[91] The gold standard comprises 41 classes and 22 relations and is based on the domains *people*, *media*, *organizations*, *geography*, and *biology*, as introduced in Section 4.1.2. The classes are geared to corresponding WordNet synsets.

Schema Completeness The criterion *schema completeness* focuses on the completeness of the KG schema, i.e., w.r.t. its classes and relations [35].

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

[91] See http://km.aifb.kit.edu/sites/knowledge-graph-comparison/.

Table 6
Results from the completeness evaluation.

	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

Evaluation method We evaluate the schema completeness by means of the previously introduced gold standard.

Evaluation results

The evaluation results are shown in Table 6 and are discussed in the following:

DBpedia:

1. Classes: The DBpedia ontology was created manually and covers all domains well. However, it is incomplete in the details and therefore appears superficial. For instance, within the domain of plants the DBpedia ontology uses not the class "tree", but the class "ginko", which is a subclass of trees. We can mention as reason for such gaps and incorrect 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, as our evaluation shows (coverage of 0.91). Some missing relations or modeling failures are due to the Wikipedia info-box characteristics. For example, to represent the sex of a person the existing relation `foaf:gender` seems to fit. However, it is only modeled in the ontology as belonging to the class `language` and not used on instance level. The sex of the person is often not explicitly mentioned in the Wikipedia info-boxes, but implicitly mentioned in the category names (e.g., "American male singers"). While DBpedia does not exploit this knowledge, YAGO uses it and provides statements with the relation `yago:hasGender`.

YAGO: 1. Classes: To create the set of classes, the Wikipedia categories are extracted and connected to WordNet synsets. The schema completeness is already covered by the WordNet classes.

2. Relations: The YAGO schema does not contain many distinct, but rather abstract relations, which can be understood in different senses. The abstract relation names make it often difficult to infer the meaning; often the meaning of relations is given only after considering the corresponding classes (domain and range; e.g., `yago:created` in the sense of "the director of the movie"). The relation `yago:wasCreatedOnDate` can be used reasonably for both the foundation year of a company and for the publication date of a movie. We can state that expanding the YAGO schema by further relations appears reasonable.

Freebase: 1. Classes: Freebase lacks a class hierarchy and subclasses of classes are often in different namespaces (e.g., the class people `/person/people` has the subclasses artists `/music/artist` and sportsmen `/sports/pro_athlete`), which makes it difficult to find suitable subclasses and super classes. Noticeable is also that the biology domain contains no subclasses. This is due to the fact that families are represented as entities (e.g., tree `/m/07j7r` or ginko `m/0htd3`). The ginko tree is then not classified as tree, but by a generic class `/biology/oganism_classification`.

2. Relations: According to our gold standard, Freebase is "relation-complete". Since a given entity can be described by relations from different namespaces, many relations are generally available.

Wikidata: Wikidata is complete both w.r.t. to classes and relations. Besides frequently used generic classes such as "human" (`wdt:Q5`) for people also special classes exist. Interesting is also that Wikidata covers all relations of the gold standard, even though it contains considerably less relations (1,874 vs. 70,802). The Wikidata methodology to let users propose new relations, to discuss about their coverage and reach, and finally to approve or disapprove the relations, seems to be appropriate.

OpenCyc: The ontology of OpenCyc was created manually and covers both generic and specific classes such as social groups and "LandTopographicalFeature". We can measure that OpenCyc is complete w.r.t. the classes. Regarding the relations, OpenCyc lacks some relations of the gold standard such as the number of pages or the ISBN of books.

Column Completeness **Evaluation method** During creation of the gold standard, we ensured that we select only those relations to which a value typically exists (e.g., there is no death date for living people). We measure the schema completeness by calculating the average mean of all found class-relation-combinations which can occur on the instance level based on the schema information. In total, we create 25 combinations based on the gold standard.

Results Table 6 shows the results of our evaluation. DBpedia and OpenCyc score well. Despite the high

Table 7
Population Completeness regarding the different domains.

Domains	DB	FB	OC	WD	YA
People	1	1	0.45	1	1
Media	0.75	0.80	0.40	0.95	0.85
Organizations	1	1	0.35	1	1
Geography	1	1	0.80	1	1
Biology	0.90	0.90	0.40	1	0.60

number of 3M represented people, Wikidata achieves a high coverage of birth dates (70.3%) and of the sex (94.1%). YAGO obtains a coverage of 63.5% for the sex. DBpedia, in contrast, does not contain this relation in its ontology. If we consider the DBpedia relation `dbp:gender`, which originates from the generic info-box extractor, this leads to a coverage of only 0.25% (5,434 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 surprisingly shows a high coverage (92.7%) of the authors of modeled books, given the basic population of 1.7 M books. Note, however, that there are not only single books modeled under `/book/book`, but also other entries such as a description of the Lord of Rings (`m/07bz5`). ISBN are covered in 63.4% of the cases. OpenCyc breaks ranks, as it contains mainly taxonomic knowledge so that no values for the considered relations are stored in the KG.

Population Completeness **Evaluation method** For evaluating the population completeness, we reuse the gold standard introduced above. For each domain, five classes for selected and for each class two well-known entities (called "short head") and two rather unknown entities (called "long tail") were chosen based on the selection criterion described in the following. Hereby, only entities were considered which existed already in 2010, so that this criterion is measured independently from the timeliness.

The well-known entities for the different domains were chosen "world-wide" and without temporal restrictions. To take the most popular entities per domain, we used quantitative statements. For instance, to select well-known sportsmen, we ranked them by the number of won olympic medals; to select the most popular mountains, we ranked the mountains by their heights.

To select the rather unknown entities for the domains, we considered entities in the context of both Germany and a specific year. For instance, regarding the sportsmen, we selected German sportsmen 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[92].[93]

Table 8
Timelineness of KGs.

	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

Since in our gold standard each of the five domains contains five classes and since for each class two well-known and two rather unknown entities were selected, 100 entities were evaluated in total (see our website).

Evaluation results The results of our evaluation are shown in Table 6. DBpedia, Freebase, and Wikidata are complete w.r.t. well-known entities. YAGO lacks some well-known entities of our gold standard, although some of them are represented in Wikipedia. One reason for that fact is that those Wikipedia entities are not imported into YAGO for which a WordNet class exists. For instance, there is no "Great White Shark" entity, only a WordNet class `yago:wordnet_great_white_shark_101484850`.

Not very surprising is the fact that all KGs show a higher degree of completeness regarding well-known entities than regarding rather unknown entities. The reason for that is that general knowledge, which the considered KGs want to capture, mainly covers well-known entities.

Noteworthy is in particular the high population completeness degree for Wikidata also 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 automatically 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 often English labels for the entities are missing.

4.2.6. Timeliness

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

[93] 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.

Timeliness frequency of the KG Wikidata provides the highest fulfillment score for this criterion. Modifications in Wikidata are via browser and via HTTP URI dereferencing immediately visible. Hence, Wikidata falls in the category of continuous timeliness. Besides that, an RDF export is provided on a roughly monthly basis. The DBpedia KG is created about once a year and it is not modified in the meantime. In the last 36 months (as of February 2016), three DBpedia versions have been published (April 2015, September 2014, and September 2013).[94] Besides the static DBpedia, the DBpedia live[95] is continuously updated by tracking changes in Wikipedia in real-time.[96] OpenCyc and YAGO have been updated less than once per year. The last OpenCyc version dates from May 2012. YAGO3 was published 2015, YAGO2 in 2011 and the interim version YAGO2s in 2013. Both KGs are developed further, but no exact dates of the next version are known. Freebase had been updated continously until its close-down in March 2015.[97]

Specification of the validity period of statements In YAGO, Freebase, and Wikidata the temporal validity period of statements (e.g., the term in office of a president) can be specified (see the values for $m_{Validity}$ in Table 8). In YAGO, this is done via the relations `yago:occursSince`, `yago:occursUntil`, and `yago:occursOnDate`; in Wikidata, via relations "start time" (`wdt:P580`) and "end time" (`wdt:P582`). In Freebase, Compound Value types (CVTs) are used to represent higher-degree relations [37] such as `/government/government_position_held`.

Specification of the modification date of statements Freebase keeps the modification dates of statements in the KG: Via the relation `freebase:freebase/valuenotation/is_reviewed` the date of the last review of facts can be represented. Noteworthy is that in the DBpedia ontology the attribute `dcterms:modified` is used to state the date of the last revision of the DBpedia ontology. In case of Wikidata HTTP URI dereferencing, the latest modification date of a *resource* (and not of a statement) is returned via `schema:dateModified`. Also, the date of the last verification can be returned.

Table 9
Comprehensibility of KGs.

	DB	FB	OC	WD	YA
m_{Desc}	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

4.2.7. Ease of Understanding

Description of resources **Evaluation method.** We measured the extent to which resources are described per KG by submitting SPARQL requests. 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 uses `freebase:common.topic.description`.[98]

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.[99]

Surprisingly, DBpedia shows a relatively low coverage w.r.t. labels and descriptions (only 70.4%). Our manual investigations suggest that statements of higher degrees are modeled by means of intermediate nodes which have no labels, so that reification is not necessary.[100] Due to the self-describing URIs of DBpedia (URIs are derived from the titles of the English Wikipedia articles), the meaning of DBpedia resources is admittedly mostly obvious.

[94] The online version for browsing and HTTP lookups has always been the latest RDF dump version.

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

[96] However, DBpedia live does not provide the full range of relations as DBpedia.

[97] See `https://plus.google.com/109936836907132434202/posts/bu3z2wVqcQc`, requested Mar 4, 2016.

[98] Human-readable resource descriptions may also be represented by other relations [16]. However, we focused on those relations which are commonly used in the considered KGs.

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

[100] E.g., `http://dbpedia.org/page/Nayim` links via `dbo:careerStation` to 10 entities of his carrier stations.

Labels in multiple languages **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 the literal values such as "@de".

Evaluation results. According to this evaluation method, 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.

We also measured the coverage of selected languages in the KGs by means of the language annotations.[101] This was done for the languages English, French, German, Spanish, and Italian. Our results show that DBpedia, YAGO, and Freebase achieve a high coverage regarding the English language. However, only YAGO has a coverage of over 10 % for German. In contrast to those KGs, Wikidata shows a 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 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 provides RDF/XML, which is regarded as not easily understandable. We provide an overview of all provided serialization formats of the different KGs in Section 4.2.8.

Self-describing URIs We can see here two different paradigms of URI usage: On the one hand, DBpedia, OpenCyc, and YAGO rely on human-readable URIs, and therefore achieve the full fulfillment score. 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 generic and one using self-describing URIs.

Table 10
Interoperability of KGs.

	DB	FB	OC	WD	YA
m_{Reif}	1	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

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 ids: Wikidata uses Q-IDs for resources and P-IDs for relations. Freebase uses M-IDs for identifying resources. Entities can be alternatively identified by self-describing keys (e.g., `/en/Michael_Jackson` instead of `/m/09889g`). In the RDF export, those self-decribing keys are only provided as literals of the relation `/key/en` (or in other languages with the corresponding language code). Classes and relations are identified via self-describing URIs.[102]

4.2.8. Interoperability

Avoiding blank nodes and RDF reification RDF reification allows to represent further information about single RDF triples. Wikidata makes extensive use of RDF reification. Due to that, SPARQL queries for Wikidata data are more complex than SPARQL queries for KGs without reification. YAGO and Freebase also uses reification, e.g., in order to store the provenance information. However, most of the statements are usable without reification.

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

Provisioning of several serialization formats DBpedia, YAGO, and Wikidata fulfill the criterion of provisioning several RDF serialization formats to the full extent, as they provide data during the URI dereferencing alternatively in RDF/XML or other serialization formats. DBpedia and YAGO provide (besides N3/Turtle, N-Triples and JSON) further RDF serialization formats via their SPARQL endpoints, such as JSON-LD, Microdata, and CSV.

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

[102] E.g., `/music/album` for the class music alums and `/people/person/date_of_birth` for the relation birth day of a person.

Freebase is the only KG providing RDF only in Turtle format.

Using external vocabulary This criterion indicates how often external vocabulary is used in comparison to proprietary vocabulary. For that, for each KG we divide the occurrence number of triples with external relations by the number of all relations in this KG.

DBpedia uses 37 distinct external relations from 8 different vocabularies, while the other KGs mainly restrict themselves to the external vocabularies of RDF, RDFS, and OWL. In accordance with that, DBpedia is covered by external vocabulary to a large extent (ratio of 0.610).

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

Interoperability of proprietary vocabulary **Evaluation method.** This criterion determines the degree to which URIs of proprietary vocabulary are linked to external vocabulary via equivalency relations. For each KG, we measure which classes and relations are linked via `owl:sameAs`,[103] `owl:equivalentClass` (in Wikidata: `wdt: P1709`), and `owl:equivalentProperty` (in Wikidata `wdt:P1628`) to external vocabulary. Although other relations such as `rdf:subPropertyOf` could be taken into account; however, in this work we only consider equivalency relations. Regarding DBpedia, we only consider the DBpedia ontology. Freebase only provides `owl:sameAs` links in the form of a separate RDF file, but these links are only on instance level. YAGO contains 553.768 `owl:equivalentClass` links to classes under the namespace `http://dbpedia.org/class/yago`. However, as the YAGO class hierarchy was imported into DBpedia, we do not count these links as external links for YAGO.

Regarding its classes, DBpedia reaches a relative high interlinking degree of about 48.4%, using FOAF, Wikidata, Schema.org and DUL[104] as linking targets.

[103] As OpenCyc contains `owl:sameAs` also on the schema level, we consider this relation also here.

[104] See `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl`.

Table 11
Accessibility of KGs.

	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_{HTML_RDF}	1	1	1	1	0
m_{Meta}	1	0	0	0	1

Regarding its relations, DBpedia links to Wikidata and schema.org,[105] but reaches only a relation linking coverage of 6.3%. The reason for that lies in the fact that many of the relations in the DBpedia ontology are not or rarly used (see Section 4.1.3).

Regarding the classes, Wikidata provides links mainly to DBpedia. Considering all Wikidata classes, only about 0.1% of all Wikidata classes are linked to equivalent external classes. This may be the result of the high number of classes in Wikidata in general. Regarding the relations, Wikidata provides links in particular to FOAF and schema.org relations and achieves here a linking coverage of 2.1%. Although this is low, important relations are linked.[106]

In OpenCyc, about half of all classes exhibit at least one external linking via `owl:sameAs`.[107] 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, DBpdia, 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, for instance, Medelyan et al. [33]).

4.2.9. Accessibility

Dereferencing possibility of resources **Evaluation method** We measured the dereferencing possibilities

[105] Examples are `dbo:birthDate` linked to "date of birth" (`wdt:P569` and `schema:birthDate`.

[106] Such as `rdf:type` and `rdfs:subClassOf`.

[107] OpenCyc makes not difference between instance and schema level for `owl:sameAs`. The OWL primer states "The built-in OWL property owl:sameAs links an individual to an individual", but also "The owl:sameAs statements are often used in defining mappings between ontologies", see `https://www.w3.org/TR/owl-ref/#sameAs-def`.

of resources by means of trying to dereference URIs containing the fully-qualified domain name of the KG. For that, we randomly selected 15,000 URIs in the subject, predicate, and object position using the SPARQL option "ORDER BY RAND()". We submitted HTTP requests with the HTTP accept header field set to `application/rdf+xml` in order to perform content negotiation.

Evaluation results

DBpedia and OpenCyc dereferenced all URIs successfully and returned appropriate RDF data, so that they fulfilled this criterion completely. For DBpedia, 45,000 URIs wer analysed, for OpenCyc only 30,116 due to the small number of distinct predicates. Almost the same picture for YAGO: No notable errors during dereferencing could be measured.

For Wikidata, which contains also not so many unique predicates, we could analyse 34,812 URIs. URIs of Wikidata relations with supplements (e.g. the supplement "s" as in `wdt:P1024s` is used for relations referring to a statement), as they were recently introduced, could not be dereferenced. Furthermore, the blank nodes for reification on the subject and object position cannot be dereferenced.

Regarding Freebase, mainly all URIs on subject and object position of triples could be dereferenced. A few resources were not dereferencable repeatedly (HTTP server error 503; e.g., `m/0156q`). Interestingly, server errors also appear while browsing Freebase, so that data is partially not available.[108] Regarding the predicate position, many URIs are not dereferencable due to server errors (503) or due to unknown URIs (404) .[109] If a large number of Google API requests are performed (as in case of Freebase requests), an API key is necessary. In our experiments, the access was blocked after a few thousand requests. The API key would increase the daily limit up to 100,000 requests. In conclusion, without an API key, the Freebase KG is usable in limited form only.

Availability of the KG **Evaluation method** We measured the availability of KGs with the monitoring web service Pingdom.[110] For each KG, an uptime test was set up, which checks the availability of a certain resource for successful URI dereferencing (i.e., returning HTTP status code 200 OK) every minute over the time range of 60 days (Dec 18, 2015–Feb 15, 2016).

Evaluation result The online version of YAGO showed the worst availability performance. Here, the outage was on a regular basis and lasted long (see the diagram on our website).

While the other KGs showed almost no outages and were again online after some minutes on average, YAGO was on average 3.5 hours offline per outage. 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 The SPARQL endpoints of DBpedia and YAGO are provided by a Virtuoso server, the Wikidata SPARQL endpoint via Blazegraph. Freebase and OpenCyc do not provide an official SPARQL endpoint. However, an endpoint for the MQL query language for the Freebase KG is available.

Especially regarding the Wikidata SPARQL endpoint we observed tender limitations: The maximum execution time per query is set to 30 seconds; there is no limitation regarding the returning number of rows, but the frontend of the endpoint crashed in case of large result sets.[111] 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 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 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 current criterion of providing content negotiation.

The endpoints for DBpedia[112], Wikidata[113], and YAGO[114] correctly return the appropriate RDF serial-

[108] See `http://www.freebase.com/m/0156q`, the page about "Berlin", requested on Mar 5, 2016 and Apr 7, 2016.

[109] Besides that, a URL forwarding is performed from `http://rdf.freebase.com/ns/m.03hrz` to `http://www.googleapis.com/freebase/v1/rdf`.

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

[111] Querying up to 1.5M rows was possible in our experiments.

[112] `http://dbpedia.org/resource/[resource]`

[113] `https://www.wikidata.org/entity/[Resource]`

[114] `http://www.yago-knowledge.org/resource/[resource]`

Table 12
Provisioning machine-readable licensing information of KGs.

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

ization format or the corresponding HTML representation. Freebase[115], however, currently does not provide any content negotiation or HTML resource representation. Currently, only `text/plain` is returned as content type.

Noteworthy is also that regarding the N-Triples serialization YAGO and DBpedia expect the accept header `text/plain` and not `application/n-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.

Linking HTML sites with RDF serialization All KGs except OpenCyc interlink the HTML representations of resources with the corresponding RDF representations by means of `<link rel="alternate" type="[content type]" href="[URL]" title=" " />` in the HTML headers.

Provisioning of metadata about the KG For this criterion we measured which KG metadata is available (such as in the form of an VoID file[116]). DBpedia integrates the VoID vocabulary directly in its KG[117] 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.

4.2.10. License

Provisioning machine-readable licensing information DBpedia and Wikidata provide licensing information about their KG data in machine-readable form. For DBpedia, this is done in the ontology via `cc:license`[118] and links to either CC-BY-SA[119] or GNU Free Documentation License (GNU FDL)[120]. Wikidata embeds licensing information during the derefrencing of resources in the RDF document via `cc:license` and a link to the CC0 license.[121] YAGO and Freebase do not provide machine-readable licensing information. However, their data is published under the CC-BY license.[122] OpenCyc embeds licensing information into the RDF document during dereferencing, but not in machine-readable form (only as plaintext with further information under `rdfs:comment`).

Table 13
Linking via *owl:sameAs* of KGs.

	DB	FB	OC	WD	YA
m_{Inst}	0.59	0.02	0.44	0 (.65)	0.31
m_{URIs}	0.93	0.95	0.89	0.96	0.96

4.2.11. Interlinking

Linking via owl:sameAs For this metric, we queried all subjects of `owl:sameAs` triples in each KG, where the resource in the object position is out of the domain of the KG (i.e., an "external" resource). In case of Wikidata, we in addition used the stored data source identifiers such as the Freebase identifier `wdt:P646`. The reason for that is that Wikidata does not provide any `owl:sameAs` links, but that instead identical entities in other data sources are stored via these identifiers. I.e., `owl:sameAs` links can be created via URI patterns.

DBpedia and Wikidata achieved the best results w.r.t. this metric. In DBpedia, there are about 12M instances with at least on `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, 59.2% of the instances have 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 Wikidata, no `owl:sameAs` links are provided, and also no corresponding proprietary relation is available. Instead, Wikidata uses proprietary relations for instance equivalencies. Identifiers are instances of the class "Wikidata property representing a unique identifier" (`wdt:Q19847637`). The M-ID of a Freebase instance is then stored via the relation "Freebase identifier" (`wdt:P646`) as literal value (e.g.,

[115] `http://rdf.freebase.com/ns/[resource]`
[116] See `https://www.w3.org/TR/void/`, requested Apr 7, 2016.
[117] See `http://dbpedia.org/void/page/Dataset`, requested on Mar 5, 2016.
[118] Using namespace `http://creativecomons.org/ns#`.
[119] `http://creativecomons.org/licenses/by-sa/3.0/`
[120] `http://www.gnu.org/copyleft/fdl.html`.
[121] See `http://creativecomons.org/publicdomain/zero/1.0/`.
[122] See `http://createivecommons.org/licenses/by/3.0`

"/m/01x3gpk"). So far, 426 distinct identifiers are maintained this way. The identifiers need to be transformed into valid URIs during the RDF export. The Browser interface of Wikidata already transforms the identifiers. Counting at most one identifier per resource, we obtain 12,151,147 resources and, hence, a coverage of 65%. However, although the links provide relevant contents, not always an RDF representation of the resources are available; instead, the representation is often in HTML.

Validity of external URIs Regarding the dimension *accessibility*, we already analyzed the dereferencing possibility of resources using KG namespace. Now we analyse the external links of the KG. This includes `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).

The external links are valid in most cases for all KGs: All KGs obtain a metric value between 0.89 and 0.96. OpenCyc contains mainly external links to non-RDF-based Web resources to wikipedia.org and w3.org. Despite a few invalid links, the links are valid. Regarding the `owl:sameAs` links, YAGO and Freebase achieve high metric values. This is due to the fact that YAGO links mainly to DBpedia and GeoNames and Freebase mainly to Wikidata. The corresponding resources are highly available there.

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 95.5% of the 2,451 URIs without errors.

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

Noticeable is that DBpedia and OpenCyc contain many `owl:sameAs` links to domains which do not exist anymore (e.g., `http://rdfabout.com`, `http://www4.wiwiss.fu-berlin.de/factbook/`, and `http://wikicompany.org`). One solution for such invalid links might be to remove them if they have been invalid for a certain time span.

4.3. Bottom Line

We now summarize the evaluation results of Section 4.2:

- *Syntactic Validity of RDF documents* All KGs provide syntactic valid RDF documents.
- *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 500K 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 triples low. In case of Wikidata, some few invalid literals such as the ISBN have been corrected since the time of evaluation. This indicates that knowledge in Wikidata is curated continuously.
  For DBpedia, comments next to the values to be extracted (such as ISBN) in the info-boxes of Wikipedia led to inaccurate extracted values.
- *Semantic Validity of Triples* All considered KGs scored well regarding this metric. Note, however, that a qualitatively better evaluation is achievable by a manual evaluation.
- *Trustworthiness on KG level* Based on the way of how data is imported and curated, OpenCyc and Wikidata can be trusted the most.
- *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 1.3% of the statements have provenance information attached to them. Note, however, that not every statement in Wikidata requires a reference and that it is hard to evaluate which statements lack such a reference.
- *Using unknown and empty values* Wikidata and Freebase allow the indication of empty values, Wikidata also allows storing which values are unknown.
- *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.
- *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.

- *Consistency of statements w.r.t. relation constraints*
  The inconsistencies of all KGs regarding the range indications of relations are mainly due to inconsistent data types and due to missing instantiations of the instances. In order to achieve a higher instantiation ratio, instances could be instantiated by the respective classes as soon as those resources occur in relations where domain and, respective, range are defined.
  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.
- *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.
- *Schema Completeness* The DBpedia ontology was created manually and covers all domains well. However, it is incomplete in many details. The relations are considerably well covered in the DBpedia ontology.
  The YAGO classes are connected to WordNet synsets. Some YAGO relations (e.g. `yago:created`) are ambiguous and can therefore be understood in different senses.
  Freebase does not use a class hierarchy and classes are grouped into different domain, i.e. it is sometimes difficult to find related classes if they are not in the same domain. The Freebase relations are complete w.r.t. our gold standard, since they are widely applicable and since many of them are available.
  Wikidata covers all relations of the gold standard, even though it contains considerably fewer relations than Freebase for instance. This is due to the Wikidata process of approving relations before adopting them.
  OpenCyc is complete w.r.t. the classes; it lacks some relations of the gold standard.
- *Column Completeness* DBpedia and OpenCyc show the best column completeness values, i.e. many entities have values for relations which are defined on the schema level.
- *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.
- *Timeliness frequency of the KG* Only Wikidata provides the highest fulfillment score for this criterion, as it provided continuous timeliness for URI dereferencing and an RDF export on a monthly basis.
- *Specification of the validity period of statements* In YAGO, Freebase, and Wikidata the temporal validity period of statements (e.g., term in office) can be specified.
- *Specification of the modification date of statements* Only Freebase keeps the modification dates of statements. Wikidata provides the modification date during URI dereferencing.
- *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 especially statements of higher degrees such as career stations of people are not modeled via blank nodes, but via intermediate nodes for which no labels are provided.
- *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, in contrast, does not have such a high coverage regarding English, but instead covers other languages considerably. 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.
- *Understandable RDF serialization* DBpedia, YAGO, and Wikidata provide several understandable RDF serialization formats. Freebase only provides the understandable format RDF/Turtle. OpenCyc relies only on RDF/XML, which is seen as not easily readable for humans.
- *Self-describing URIs* We can find mixed paradigms regarding the URI generation: DBpedia, YAGO, and OpenCyc rely on human-readable URIs, while Wikidata and Freebase (in part, i.e. for resources) use identifiers.
- *Avoiding blank nodes and RDF reification* Wikidata, YAGO and Freebase are the KGs which use reification. This is important when querying and reusing the data.

- *Provisioning of several serialization formats* Freebase is the only KG providing data in the serialization format RDF/Turtle only.
- *Using external vocabulary* DBpedia and Wikidata show high degrees of external vocabulary usage. Regarding DBpedia, the RDF, RDFS, and OWL vocabulary is used. Wikidata has a high external vocabulary ratio, since there are many language labels and descriptions (modeled via `rdfs:label` and `schema:description`). Also, due to instantiations of statements for reification purposes, `rdf:type` is used a lot.
- *Interoperability of proprietary vocabulary* While almost every second class in DBpedia is linked to external classes, only 6.3% of all relations have links to external relations, i.e. relations not withing the DBpedia namespace. Note, that many of the DBpedia relations are not or rarely used.
  Wikidata shows a very low interlinking degree of classes to external classes and of relations to external relations. As the introduction of new relations needs to be approved, classes can be immediately introduced. We see, however, no influence of this fact in the number of links to external classes and relations.
  Surprising is that half of all OpenCyc classes exhibit at least one `owl:sameAs` link.
- *Dereferencing possibility of resources* Resources in DBpedia, OpenCyc, and YAGO can be dereferenced without considerable issues. Wikidata has introduced relations with supplements and also uses blank nodes. Those kinds of terms are not dereferencable. For Freebase we measured a quite considerable amount of dereferencing failures due to server errors and unknown URIs. Note also that Freebase requires an API key for many API requests.
- *Availability of the KG* While all other KGs showed almost no outages, YAGO shows a noteworthy instability regarding its online availability. We measured 109 outages for YAGO in the given time interval, taking on average 3.5 hours.
- *Provisioning of public SPARQL endpoints* Noteworthy is here that the Wikidata SPARQL endpoint has a maximum execution time per query of 30 seconds. This might be a bottleneck for some queries.
- *Provisioning of an RDF export* Mostly, RDF export data of the KGs is provided in N-Triples and the Turtle format.
- *Support of content negotiation* OpenCyc does not support any content negotiation; only RDF/XML is provided. Freebase currently only returns `text/plain` as content type.
- *Linking HTML sites with RDF serialization* All KGs except OpenCyc interlink the HTML representations of resources with the corresponding RDF representations.
- *Provisioning of metadata about a KG* 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.
- *Provisioning machine-readable licensing information* Only DBpedia and Wikidata provide licensing information about their KG data in machine-readable form.
- *Interlinking via `owl:sameAs`* DBpedia and Wikidata provide the highest number of instance equivalency relations (`owl:sameAs` links in DBpedia and the proprietary relations with literals in Wikidata). Based on the resource interlinkage, these KGs may be called Linked Data hubs among the considered KGs.
- *Validity of external URIs* The links to external Web resources are valid in most cases for all KGs. OpenCyc contains mainly external links to non-RDF-based Web resources at Wikipedia and w3.org. DBpedia and OpenCyc contain many `owl:sameAs` links to domains which do not exist anymore; those links could be deleted.

5. KG Recommendation Framework

We now propose a framework for selecting the most suitable KG for a given setting. The goal of using the framework is to obtain a concrete recommendation or an appropriate preselection of a given set of knowledge graphs $G = \{g_1, ..., g_n\}$. For that, the user needs to go through the steps as depicted in Figure 10.

In Step 1, the preselection criteria and the weights for the criteria are specified. The preselection criteria can be both quality criteria and 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 is an example for a general criterion. If the active curation of the KG is a criterion given from the requirements' analysis, Freebase can be excluded, since this KG is not curated anymore.

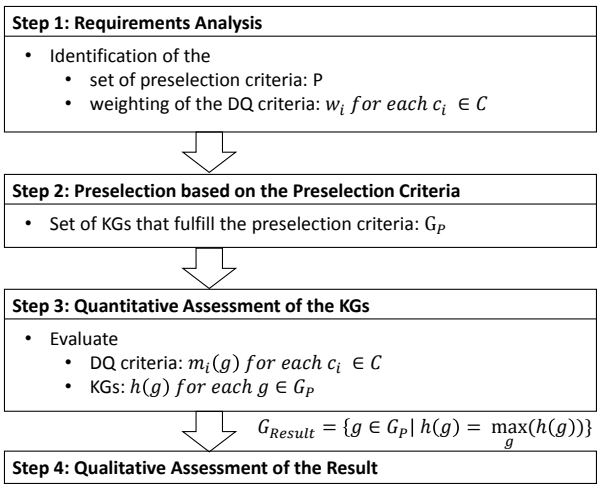


Fig. 10. Proposed Selection Process when using our framework.

After weighting the criteria and dimensions, those KGs are neglected which do not fulfill the preselection criteria.

Afterwords, the fulfillment degrees of the remaining KGs are calculated and the KG with the highest fulfillment degree is selected.

Finally, in a last step the result can be assessed w.r.t. qualitative aspects (considering our quantitative assessments of the KGs in Section 4.2) and, if necessary, another KG can be selected for the actual setting.

Table 14 shows an example of how to use our framework for ranking KGs: Given the values of all metrics for the KGs as determined in our evaluation (see Section 4.2), we can calculate a total score for each KG. This total score can either be based on an unweighted averaging of the single metric values per KG, or be based on a weighting as specified by the user (see last column in the table). Based on the total scores, the user can either take the KG with the highest score, or make a qualitative assessment of the KGs in addition (Step 4).

6. Related Work

6.1. Linked Data Quality Criteria

Zaveri et al. [42] provide a conceptual framework for quality assessment of linked data based on the systematic review of quality criteria and metrics which are assigned to quality dimensions and categories based on the framework of Wang et al. [40]. Our framework is also based on Wang's dimensions and extended by the criteria consistency [11], licensing and interlinking [42]. Further, we reintroduce the dimensions trustworthiness and interoperability as a collective term for multiple dimensions.

As many criteria and metrics are rather abstract, we select and develop criteria w.r.t. their applicability to cross-domain knowledge graphs of the linked open data cloud. Table 15 shows an overview of the most relevant papers that were considered during literature review that have similarities with criteria in our framework stating generic guidelines (e.g. for data publishers [24]) and introducing both criteria with corresponding metrics (e.g. [18,27]) and the criteria without metrics (e.g. [35,26]). In total, 27 of 34 criteria are supported by concepts described in earlier papers. Furthermore, we introduce seven new criteria m_{graph}, m_{NoVal}, $m_{checkRstr}$, $m_{Ranking}$, m_{Freq}, $m_{Validity}$ and m_{Avai}. In the following, we give an overview of which criteria have been presented so far and in which novel contexts we apply them.

Pipino et al. introduce the criteria schema completeness, column completeness and population completeness in the context of databases [35]. We introduce metrics and apply them, to the best of our knowledge, the first time on DBpedia and other cross-domain knowledge graphs.

OntoQA [38] 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 quality criteria description of resources (m_{Descr}) and column completeness (m_{cCol}). Based on a large-scale crawl of the semantic web, Hogan et al. [26] analyze quality problems based on frequent errors in publishing data in RDF. Later, Hogan et al. [27] introduce further criteria and metrics based on linked data guidelines for data publishers [24]. Whereas Hogan et al. crawl and analyze many knowledge graphs, we analyze a selected set of knowledge graphs in more detail.

Heath et al. [24] 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, e.g. „Do you refer to additional access methods“ leads to the criteria provisioning of public SPARQL endpoints (m_{SPARQL}) and provisioning of an RDF export (m_{Export}) or „Do you map proprietary vocabulary terms to other vocabularies?“ leads to the criteria interoperability of proprietary vocabulary ($m_{propVoc}$). Similar metrics to our framework that are based on

Table 14
Framework with an example weighting which would be reasonable for a user setting as given in [30].

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
	$m_{semTriple}$	1	1	1	1	1	1
Trustworthiness	m_{graph}	0.5	0.5	1	0.75	0.25	1
	m_{fact}	0.5	1	0	1	1	2
	m_{NoVal}	0	1	0	1	0	1
Consistency	$m_{checkRestr}$	0	1	0	1	0	1
	$m_{conClass}$	0.875	1	0.999	1	0.333	1
	$m_{conRelat}$	0.991	0.45	1	0	0.992	1
Relevancy	$m_{Ranking}$	0	0	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	1
	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	1
	m_{Change}	0	1	0	0	0	1
Ease of understanding	m_{Descr}	0.704	0.972	1	0.9999	1	3
	m_{Lang}	1	1	0	1	1	2
	m_{uSer}	1	1	0	1	1	1
	m_{uURI}	1	0.5	1	0	1	2
Interoperability	m_{Reif}	1	0.5	0.5	0	0.5	1
	$m_{iSerial}$	1	0	0.5	1	1	2
	m_{extVoc}	0.61	0.108	0.415	0.682	0.134	2
	$m_{propVoc}$	0.15	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	1
	m_{HTML_RDF}	1	1	0	1	1	0
	m_{Meta}	1	0	1	0	0	1
Licensing	$m_{macLicense}$	1	0	0	1	0	1
Interlinking	m_{Inst}	0.592	0.018	0.443	0	0.305	2
	m_{URIs}	0.929	0.954	0.894	0.957	0.956	1
Unweighted Average		0.708	0.605	0.498	0.738	0.625	
Weighted Average		0.718	0.575	0.516	0.742	0.646	

Table 15
Overview of Related Work regarding Data Quality Criteria for KGs.

Criterion	[35]	[38]	[26]	[24]	[18]	[20]	[27]	[41]	[2]	[31]
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_{HTML_RDF}				✓						
m_{Meta}				✓	✓		✓			
$m_{macLicense}$			✓	✓		✓				
m_{Inst}			✓			✓	✓			
m_{URIs}					✓			✓		

these guidelines can also be found in other frameworks ([27,18]).

Flemming [18] introduces a framework for the quality assessment of linked data quality that measures the linked data quality based on a sample of a few RDF documents through dereferencing. Based on a systematic literature review, criteria and metrics are introduced. Noteable, 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, we evaluate the whole knowledge graph.

SWIQA [20] is a quality assessment framework that introduces criteria and metrics for the dimensions accuracy, completeness, timeliness and uniqueness. Noteable, the dimensions accuracy is described by the criteria syntactic and sematic validity according to Batini et al. [6] and the dimension completeness is described by schema completeness, column completeness and population completeness according to Pipino et al. [35].

TripleCheckMate [2] is a framework for linked data quality assessment using a crowdsourcing-approach for manual validation of facts. Based on this approach,

Zaveri et al. [41] and Acosta et al. [3] analyze both syntactic and semantic accuracy as well as the consistency of the data of DBpedia.

Kontokostas et al. [31] present a framework for test-driven evaluation of linked data quality that is inspired by the paradigm of test-driven software development. The framework introduces 17 SPARQL templates of tests that can be used for analyzing knowledge graphs w.r.t. accuracy and consistency. Noteable, tests can also evaluate external constraints that exists due to the usage of external vocabulary, e.g. `foaf:mbox` is an inversfunctional relation. The framework is evaluated on a set of knowledge graphs including DBpedia.

6.2. Comparing KGs by Key Statistics

Duan et al [15], Tartir [38], and Hassanzadeh [23] can be mentioned as the most similar related work regarding the evaluation of the key statistics as presented in 4.1.

Duan et al. [15] analyze the structuredness of RDF data sets and describes them by simple statistical figures that are calculated based on RDF dumps in N-triple serialization. In contrast to that, we use SPARQL queries to obtain the figures, thus not limiting ourselves to the N-Tripel serialization of knowledge graphs. Duan et al. claim that simple statistical figures are not sufficient to analyze the structuredness and differences of RDF data sets introducing a coherence metric. Accordingly, we analyze not only simple statistical figures but further analyze the data quality of knowledge graphs. Duan et al. provide statistics on DBpedia, YAGO2, UniProt, and multiple benchmark datasets.

OntoQA [38] introduces metrics that can be used for the analysis of ontologies, e.g. class richness that is defined as ratio of number of classes with and without instances. Tartir et al. do not analyze knowledge graphs but a set of ontologies, e.g. SWETO, TAP and GlycO. Noteable, the statistical measures are often not the same but at least similar, e.g. both Duan et al. and Tartir et al. calculate the ratio of instances per class whereas we consider entities as base line.

Tartir et al. [38] and Hassanzadeh et al. [23] analyze the coverage of domains. 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 as Freebase has no hierarchy. Hassanzadeh et al. analyzes the coverage of domains by listing the most frequent classes with the highest number of instances in a table. This gives only little overview of the covered domains as entities can be instances of multiple classes in the same domain, z.B. `dbo:Place` and `dbo:PopulatedPlace`. Thus, we adapt the idea of Hassanzadeh et al.: We manually map the most frequent classes to domains such as people and geography and then delete duplicate entities within a certain domain, i.e. if an entity is instantiated both as `dbo:Place` and `dbo:PopulatedPlace` the entity will be counted only once in the domain geography.

7. Conclusion

Freely available knowledge graphs (KGs) have not been in the focus of any extensive comparative study so far. In this survey, we defined 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 and process to enable readers to find the most suitable KG for their settings.

References

[1] M. Acosta, E. Simperl, F. Flöck, and M.-E. Vidal. HARE: A Hybrid SPARQL Engine to Enhance Query Answers via Crowdsourcing. In *K-CAP2015: The 8th International Conference on Knowledge Capture*. International Conference on Knowledge Capture, 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] P. Buneman, S. Khanna, and T. Wang-Chiew. Why and where: A characterization of data provenance. In *Database Theory—ICDT 2001*, pages 316–330. Springer, 2001.

[14] 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.

[15] 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 2011 ACM SIGMOD International Conference on Management of data*, pages 145–156. ACM, 2011.

[16] 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.

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

[18] 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.

[19] G. Freedman and E. Reynolds. Enriching basal reader lessons with semantic webbing. *Reading Teacher*, 33(6):677–684, 1980.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] J. Hoffart, F. Suchanek, K. Berberich, E. Kelham, G. de Melo, G. Weikum, F. Suchanek, G. Kasneci, M. Ramanath, and A. Pease. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Commun. ACM*, 52(4):56–64, 2009.

[26] 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.

[27] 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.

[28] 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.

[29] J. M. Juran, F. Gryna, and R. Bingham. *Quality Control Handbook*. McGraw-Hill, New York, 1974.

[30] 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-Verlag.

[31] 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.

[32] 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.

[33] 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.

[34] F. Naumann. *Quality-driven query answering for integrated information systems*, volume 2261. Springer Science & Business Media, 2002.

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

[36] 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-Verlag.

[37] T. P. Tanon, D. Vrandečić, S. Schaffert, T. Steiner, and L. Pintscher. From freebase to wikidata: The great migration. In *World Wide Web Conference*, 2016.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.