Semantic Web 0 (2015) 1–0
IOS Press

Automatic Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods

Heiko Paulheim,
Data and Web Science Group, University of Mannheim, B6 26, 68159 Mannheim, Germany
E-mail: heiko@informatik.uni-mannheim.de

Abstract. In the recent years, different web knowledge graphs, both free and commercial, have been created, with DBpedia, YAGO, and Freebase being among the most prominent ones. Those graphs are often constructed from semi-structured knowledge, such as Wikipedia, or harvested from the web with a combination of statistical and NLP methods. The result are large-scale knowledge graphs that try to make a good trade-off between completeness and correctness. In order to further increase the utility of knowledge graphs, various refinement methods have been proposed, which try to infer and add missing knowledge to the graph, or identify erroneous pieces of information. In this article, we provide a survey of such *knowledge graph refinement* approaches, with a dual look at both the methods being proposed as well as the evaluation methodologies used.

Keywords: Knowledge Graphs, Refinement, Completion, Error Detection, Evaluation

1. Introduction

Knowledge graphs on the web are a backbone of many information systems that require access to structured knowledge, be it domain-specific or domain-independent. The idea of feeding intelligent systems and agents with general, formalized knowledge of the world dates back to classic Artificial Intelligence research in the 1980s. More recently, with the advent of Linked Open Data sources like DBpedia [41], and by Google's announcement of the Google Knowledge Graph in 2012[1], representations of general world knowledge as graphs have drawn a lot of attention again.

There are various ways of building such knowledge graphs. They can be curated like *Cyc* [42], edited in a community-based way like *Freebase* [7] and *Wikidata* [81], or extracted from large-scale, semi-structured web knowledge bases such as Wikipedia, e.g., *DBpedia* [41] or *YAGO* [77]. Furthermore, information extraction methods for unstructured or semi-structured information are proposed, e.g., for *NELL* [11], *PROSPERA* [52], or *KnowledgeVault* [16].

In any case, knowledge graphs are never perfect [8]. As a model of the real world or a part thereof, formalized knowledge cannot reasonably reach *full coverage*, i.e., contain information about every entity in the real world. Furthermore, it is unlikely, in particular when heuristic methods are applied, that the knowledge graph is *fully correct* – there is usually a trade-off between coverage and correctness, which is addressed differently in each knowledge graph. [88]

In this survey, we review methods that have been proposed for improving existing knowledge graphs. In many cases, those methods are developed by researchers outside the organizations which *create* the knowledge graphs. They rather take an existing knowledge graph and try to increase its coverage and/or correctness by various means. Thus, the focus of this survey is not knowledge graph *construction*, but knowledge graph *refinement*.

[1] http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html

Decoupling knowledge base construction and refinement has different advantages. First, it allows for developing methods for refining arbitrary knowledge graphs, which can then be applied to improve multiple knowledge graphs. Other than fine-tuning the heuristics that create a knowledge graph, the impact of such generic refinement methods can thus be larger. Second, evaluating refinement methods in isolation of the knowledge graph construction step allows for a better understanding and a cleaner separation of effects, i.e., it facilitates more qualified statements about the effectiveness of a proposed approach.

The rest of this article is structured as follows. Section 2 gives a brief introduction into knowledge graphs in the Semantic Web. In section 3 and 4, we present a categorization of approaches and evaluation methodologies. In section 5 and 6, we present the review of methods for completion (i.e., increasing coverage) and error detection (i.e., increasing correctness) of knowledge graphs. We conclude with a critical reflection of the findings in section 7, and a summary in section 8.

2. Knowledge Graphs in the Semantic Web

From the early days, the semantic web has promoted a graph-based representation of knowledge, e.g., by pushing the RDF standard[2]. In such a graph-based knowledge representation, *entities*, which are the nodes of the graph, are connected by *relations*, which are the edges of the graph (e.g., *Shakespeare has written Hamlet*), and entities can have types, denoted by *is a* relations (e.g., *Shakespeare is a Writer*, *Hamlet is a play*). In many cases, the entity and relation types are organized in a *schema* or *ontology*, which defines their interrelations and restrictions of their usage.

With the advent of Linked Data [5], it was proposed to interlink different datasets in the semantic web. By means of interlinking, the collection of could be understood as one large, global knowledge graph (although very heterogenous in nature). To date, roughly 1,000 datasets are interlinked in the *Linked Open Data cloud*, with the majority of links connecting *identical* entities in two datasets [73].

Although, from a broader perspective and by definition, any graph-based representation of some knowledge could be considered a *knowledge graph* (this would include any kind of RDF dataset), the term is typically used to refer to larger, cross-domain datasets. Knowledge graphs can be built using different methods: they can be curated by an organization or a small, closed group of people, crowd-sourced by a large, open group of individuals, or created with heuristic, automatic or semi-automatic means.

OpenCyc is a freely available version of the *Cyc* knowledge base, which dates back to the 80s [42]. Rooted in traditional artificial intelligence research, it is a *curated* knowledge graph, developed and maintained by CyCorp Inc.[3]. A semantic web endpoint to OpenCyc also exists, containing links to DBpedia and other LOD datasets.[4]

Curating a universal knowledge graph is an endeavour which is infeasible for most individuals and organizations. To date, more than 900 person years have been invested in the creation of Cyc [71], with gaps still existing. Thus, distributing that effort on as many shoulders as possible through *crowdsourcing* is a way taken by *Freebase* and *Wikidata*[5], where a schema is usually defined by a smaller group of people (e.g., defining that each person should have a birth date), and populated by the crowd (who, in the example, fill in the actual birth dates). Freebase also serves as a seed for the non-public *Google Knowledge Graph*, which is used by Google in its search and combines data from various sources, including knowledge harvested from the social network Google+ and websites enriched with schema.org MicroData.

DBpedia and *YAGO* go a different way, relying on Wikipedia as a large-scale knowledge collection which already exists and is constantly maintained and extended by millions of users. While DBpedia relies on crowd-sourced mappings to a centralized ontology for extracting knowledge from infoboxes [41], YAGO builds its classification implicitly from the category system in Wikipedia and the lexical resource *WordNet* [49], with infobox properties manually mapped to a fixed set of attributes [47]. While DBpedia creates different interlinked knowledge graphs for each language edition of Wikipedia [10], YAGO aims at an automatic fusion of knowledge extracted from various Wikipedia language editions.

While DBpedia and YAGO use semi-structured content as a base, methods working on unstructured data

[2] http://www.w3.org/RDF/

[3] http://www.cyc.com/

[4] http://sw.opencyc.org/

[5] As of March 31st, 2015, Freebase is shut down, with the data being transferred to Wikidata. See https://plus.google.com/109936836907132434202/posts/3aYFVNf92A1

have been proposed as well. One of the earliest approaches working at web-scale was the *Never Ending Language Learning (NELL)* project [11]. The project works on a large-scale corpus of web sites and exploits a coupled process which learns text patterns corresponding type and relation assertions, as well as applies them to extract new entities and relations. The system has been running until today, continuously extending its knowledge base. While not being a genuine semantic web knowledge graph, it has been shown that the data in NELL can be exposed as semantic web data as well [89]. A similar approach is taken by Google's *Knowledge Vault*, which, in addition to plain text and web sites also exploits semi-structured information on the web, like HTML tables [16].

Table 1 lists a selection of popular knowledge graphs, as well as their characteristics. The numbers follow the latest DBpedia 2014 release[6], the YAGO3 release [47], the Freebase website[7], the NELL statistics heatmap[8], the recent version of Cyc[9], an announcement about the Google Knowledge Graph by Google[10], the Wikidata class and property browser[11] and a recent article on Wikidata [81], as well as the comparison of knowledge graphs in [16] and [55].

It can be observed that the graphs differ in the basic measures, such as the number of entities and relations, as well as in the size of the schema they use, i.e., the number of classes and relations. From these differences, it can be concluded that the knowledge graphs must differ in other characteristics as well, such as averages degrees or connectivity.

3. Categorization of Knowledge Graph Refinement Approaches

Knowledge graph refinement methods can differ along different dimensions. For this survey, we distinguish the goal of the method, i.e., completion vs. correction of the knowledge graph, the targeted kind of information, as well as the data used to run the approach.

[6] http://wiki.dbpedia.org/Datasets2014 and http://wiki.dbpedia.org/Ontology2014

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

[8] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.922.heatmap.html

[9] http://opencyc.org/

[10] http://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html

[11] http://tools.wmflabs.org/wikidata-exports/miga/?classes#

3.1. Completion vs. Error Detection

There are two main goals of knowledge graph refinement: (a) adding missing knowledge to the graph, i.e., *completion*, and (b) identifying wrong information in the graph, i.e., *error detection*. From a data quality perspective, those goals relate to the data quality dimensions *free-of-error* and *completeness* [67].

3.2. Targeted Kind of Information

Both types of approaches can be distinguished into the targeted kind of information in the knowledge graph. For example, some approaches are targeted towards completing/correcting entity type information, while others are targeted to (either specific or any) relations between entities, or literal values, such as numbers, or interlinks between different knowledge graphs. Another strand of research targets the extension of the *schema* used by the knowledge graph (i.e., the T-box), not the data (the A-box)

3.3. Internal vs. External Methods

A third distinguishing property is the data used by an approach. While *internal* approaches only use the knowledge graph itself as input, *external* methods use additional data, such as text corpora. In the widest sense, approaches making use of human knowledge, such as crowdsourcing [1] or games with a purpose [82], can also be viewed as external methods. However, they are out of the scope of this survey, since we restrict ourselves only to fully automatic approaches.

4. Categorization of Evaluation Methodologies

There are different possible ways to evaluate knowledge graph refinement. On the top level, we can distinguish methodologies that use only the knowledge graph at hand, and methodologies that use external knowledge, such as human annotation.

4.1. Knowledge Graph as Silver Standard

Methodologies in the first category usually use the given knowledge graph as a silver standard. This is usually applied to measure the performance of knowledge graph *completion* approaches, where it is analyzed how well relations in a knowledge graph can replicated by a knowledge graph completion method.

Table 1
Overview of Popular Knowledge Graphs. The table depicts the number of instances and facts; as well as the number of different types and relations defined.

Name	Instances	Facts	Entity Types	Relation Types
DBpedia	4,580,000	583,000,000	685	2,795
YAGO	4,595,906	25,946,870	488,469	77
Freebase	47,560,817	2,903,361,537	26,507	37,781
Wikidata	15,759,256	43,189,154	23,157	1,203
NELL	1,908,694	441,807	274	296
OpenCyc	118,499	2,413,894	45,153	18,526
Google Knowledge Graph	500,000,000	3,500,000,000	1,500	35,000
Knowledge Vault	45,000,000	271,000,000	1,100	4,469

The result quality is usually measured in recall, precision, and F-measure. In contrast to using human annotations, large-scale evaluations are easily possible. The silver standard method is not suitable for error detection, since it assumes the knowledge graph to be correct.

There are two variants of silver standard evaluations: in the more common ones, the entire knowledge graph is taken as input to the approach at hand, and the evaluation is then also carried out on the entire knowledge graph. As this may lead to an overfitting effect (in particular for internal methods), some works also foresee the splitting of the graph into a training and a test partition, which, however, is not as straight forward as, e.g., for propositional classification tasks [54], which is why most papers use the former method.

A problem with this approach is that the knowledge graph itself is not perfect (otherwise, it would not need refinement), thus, this evaluation method may sometimes underrate the evaluated approach. More precisely, most knowledge graphs follow the *open world assumption*, i.e., an axiom not present in the knowledge graph may or may not hold. Thus, if a completion approach correctly predicts the existence of an axiom missing in the knowledge graph, this would count as a false positive and thus lower precision.

4.2. Partial Gold Standard

Another option is to use a partial gold standard. In this methodology, a subset of graph entities or relations are selected and labeled manually. Other evaluations use external knowledge graphs and/or databases as partial gold standards.

For completion tasks, this means that all axioms that *should be there* are recorded, whereas for a correction tasks, a set of axioms in the graph is manually labeled as correct or incorrect. As for the silver standard methods, the quality of completion approaches is usually measured in recall, precision, and F-measure, whereas for correction methods, accuracy and/or area under the ROC curve (AUC) are often used alternatively or in addition.

Sourcing partial gold standards from humans can lead to high quality data (given that the knowledge graph and the ontology it uses are not overly complex), but is costly, so that those gold standards are usually small. On the other hand, exploiting other knowledge graphs based on knowledge graph interlinks may yield larger-scale gold standards, but has two sources of errors: errors in the target knowledge graph, and errors in the linkage between the two. For example, it has been reported that 20% of the interlinks between DBpedia and Freebase are incorrect [87], and that roughly half of the `owl:sameAs` links between knowledge graphs connect two things which are related, but not *exactly* the same (such as the company *Starbucks* and a particular Starbucks coffee shop) [26].

4.3. Ex Post Evaluation

For ex post evaluations, the output of a given approach is given to human judges for annotation, who then label completions or flagged errors as correct and incorrect. The quality metric is usually accuracy or precision, along with a statement about the total number of completions or errors found with the approach.

While partial gold standards can be reused for comparing different methods, this is not the case for ex post evaluations. On the other hand, post hoc evaluations may make sense in cases where the interesting class is rare. For example, when evaluating error detection methods, a sample for a partial gold standard from a high-quality graph is likely not to contain a meaning-

ful number of errors. In those cases, post-hoc methodologies are often preferred over partial gold standards.

4.4. Computational Performance

In addition to the performance w.r.t. correctness and/or completeness of results, computational performance considerations become more important as knowledge graphs become larger. Typical performance measures for this aspect are runtime measurements, as well as memory consumption.

Besides explicit measurement of computational performance, a "soft" indicator may be if an approach has been evaluated (or at least the results have been materialized) on an entire knowledge graph, or only on a subgraph. The latter is often done when applying evaluations on partial gold standard, where the respective approach is only executed on entities contained in that partial gold standard.

5. Approaches for Completion of Knowledge Graphs

Completion of knowledge graphs aims at increasing the coverage of a knowledge graph is the goal of knowledge graph completion. Depending on the target information, methods for knowledge graph completion either predict missing entities, missing types for entities, and/or missing relations that hold between entities.

In this section, we survey methods for knowledge graph completion. We distinguish internal and external methods, and further group the approaches by the underlying methods used.

5.1. Internal Methods

Internal methods use only the knowledge contained in the knowledge graph itself to predict missing information.

5.1.1. Classification

Predicting a type or class for an entity given some characteristics of the entity is a very common problem in machine learning, known as *classification*. The classification problem is *supervised*, i.e., it learns a classification model based on labeled training data, typically the set of entities in a knowledge graph (or a subset thereof) which have types attached. In machine learning, *binary* and *multi-class* prediction problems are distinguished. In the context of knowledge graphs, in particular the latter are interesting, since most knowledge graphs contain entities of more than two different types. Depending on the graph at hand, it might be worthwile distinguishing *multi-label* classification, which allows for assigning more than one class to an instance (e.g., *Arnold Schwarzenegger* being both an *Actor* and a *Politician*), and single-label classification, which only assigns one class to an instance [79].

For internal methods, the features used for classification are usually the relations which connect an entity to other entities [64,69], i.e., they are a variant of *link-based classification* problems [24]. For example, an entity which has an *director* relation is likely to be a *Movie*. One of the first classification-based approaches has been proposed by Neville and Jensen [54]. The authors use Bayesian classifiers on propositional representations of the graph entities. In a more recent work by Sleeman and Finin [75], the use of Support Vector Machines (SVMs) has been proposed to type entities in DBpedia and Freebase. The authors also exploit interlinks between the knowledge graphs and classify instances in one knowledge graph based on properties present in the other, in order to increase coverage and precision. Nickel et al. [56] propose the use of *matrix factorization* to predict entity types in YAGO.

Since many knowledge graphs come with a class hierarchy, e.g., defined in a formal ontology, the type prediction problem could also be understood as a *hierarchical classification* problem. Despite a larger body of work existing on methods for hierarchical classification [74], there are, to the best of our knowledge, no applications of those methods to knowledge graph completion.

Socher et al. [76] use classification to predict the existence of a relation between two entities, i.e., the classification problem uses *pairs of entities* as instances and types of relations as classes. The train a tensor neural network to predict relations based on chains of other relations, e.g., if a person is born in a city in *Germany*, then the approach can predict that the nationality of that person is *German*. The approach is applied to Freebase and WordNet.

5.1.2. Probabilistic and Statistical Methods

Analyzing a knowledge graph can reveal probabilities of certain patterns to exist. For example, the probability of a node being of type *Actor* is high if there are ingoing edges of type *cast*. Such probabilities are exploited by the *SDType* algorithm [62,63], which is currently deployed for DBpedia and adds around 3.4

million additional type statements to the knowledge graph. Similarly, *ProSWIP* [32] uses relations to predict entity types.

On the schema level, similar statistical methods can be taken to enrich a schema with additional domains and ranges of relations, or disjointness axioms, by analyzing the co-occurence of classes and properties [78].

A similar approach is taken by Oren et al. [58] – the authors propose to predict the existence of a relation based on other relations using probabilities of co-occurence. While the authors report good results, the approach is not directly applicable for knowledge graph completion, since it only predicts the *presence* of an edge in the graph, but not the entity to connect the edge to.

In data mining, association rule mining [29] is a method that analyzes the co-occurence of items in itemsets and derives association rules from those co-occurences. For predicting missing information in knowledge graphs, those methods can be exploited, e.g., in the presence of redundant information. For example, in DBpedia, different type systems (i.e., the DBpedia ontology and YAGO, among others) are used in parallel, which are populated with different methods (Wikipedia infoboxes and categories, respectively). This ensures both enough overlap to learn suitable association rules, as well as a number of entities that only have a type in one of the systems, to which the rules can be applied. In [60], we exploit such association rules to predict missing types in DBpedia based on those redundancies.

Association rule mining can also be applied to extend schemas for knowledge graphs [80]. Here, subsumptions as well as domain and range restrictions can be learned from the co-occurence of types and relations. The approach is extended to extending the schema with disjointess axioms as well in [20].

In [22], the use of association rule mining to find property chains (e.g., the grandfather of X is the father of X's mother or father). Those chains are then applied to infer new axioms in the knowledge graph.

5.2. External Methods

External methods use sources of knowledge – such as text corpora or other knowledge graphs – which are not part of the knowledge graph itself. Those external sources can be linked from the knowledge graph, such as knowledge graph interlinks or links to web pages, e.g., Wikipedia pages describing an entity, or exist without any relation to the knowledge graph at hand, such as large text corpora.

5.2.1. Classification

For type prediction, there are also classification methods that use external data. In contrast to the internal classification methods described above, external data is used to create a feature representation of an entity.

Nuzzolese et al. [57] propose the usage of the Wikipedia link graph to predict types in a knowledge graph using a k-nearest neighbors classifier. Given that a knowledge graph contains links to Wikipedia, interlinks between Wikipedia pages are exploited to create feature vectors, e.g., based on the categories of the related pages. Since links between Wikipedia pages are not constrained, there are typically more interlinks between Wikipedia pages than between the corresponding entities in the knowledge graph.

Apriosio et al. [3] use types of entities in different DBpedia language editions (each of which can be understood as a knowledge graph connected to the others) as features for predicting missing types. The authors use a k-NN classifier with different different distance measures (i.e., kernel functions), such as overlap in article categories. In their setting, a combination of different distance measures is reported to provide the best results.

5.2.2. NLP-based methods

Text-based methods have been proposed both for predicting types and relations. Approaches for predicting types often use abstracts in DBpedia to extract definitionary clauses, e.g., using Hearst patterns [28]. Such approaches have been proposed by Gangemi et al. [23] and Kliegr [36], where the latter uses abstracts in the different languages in order to increase coverage and precision. Lange et al. [38] learn patterns on Wikipedia abstracts using Conditional Random Fields [37]. A similar approach, but on entire Wikipedia articles, is proposed by [86].[12]

The prediction of a relation between two entities is usually accomplished by *distant supervision*. Typically, such approaches use large text corpora. As a first step, entites in the knowledge graph are linked to the text corpus by means of Named Entity Recognition [31,70]. Then, based on the relations in the knowledge graph, those approaches seek for text pattern which

[12] Although both approaches do not explicitly mention DBpedia, but aim at completing missing key-value pairs in infoboxes, this can be directly transferred to extending DBpedia.

correspond to relation types (such as: *Y's book X* being a pattern for the relation *author* holding between *X* and *Y*), and then apply those patterns to find additional relations in the text corpus. Such methods have been proposed by Mintz et al. [50] for Freebase, and by Aprosio et al. [4] for DBpedia. In both cases, Wikipedia is used as a text corpus.

West et al. [84] propose the use of web search engines to fill gaps in knowledge graphs. Like in the works discussed above, they first discover lexicalizations for relations. Then, they use those lexicalizations to formulate search engine queries for filling missing relation values. Thus, they use the whole Web as a corpus, and combine information retrieval and extraction for knowledge graph completion.

5.2.3. Information Extraction from Semi-Structured Data

While text is unstructured, some approaches have been proposed that use *semi-structured* data for completing knowledge graphs. In particular, approaches leveraging on structured data in Wikipedia are found in the literature. Those are most often used together with DBpedia, so that there are already links between the entities and the corpus of background knowledge, i.e., no Named Entity Recognition has to be performed, in contrast to the distant supervision approaches discussed above.

Muñoz et al. [51] propose extraction from tables in Wikipedia. They argue that for two entities cooccuring in a Wikipedia table, it is likely that the corresponding entities should share an edge in the knowledge graph. To fill in those edges, they first extract a set of candidates from the tables, using all possible relations that hold between at least one pair of entities in two columns. Then, based on a labeled subset of that extraction, they apply classification using various features to identify those relations that should *actually* hold in the knowledge graph.

In [65], we have proposed the use of list pages in Wikipedia for generating both type and relation assertions in knowledge graphs, based on statistical methods. The idea is that entities appear together in list pages for a reason, which should be able to identify for the majority of instances. For example, instances linked from the page *List of Jewish-American Writers* should all be typed as *Writer* and include an edge *religion* to *Jewish*, as well as an edge *nationality* to *United States of America*. Once such patterns are found for the majority of the list items, they can be applied to the remaining ones to fill gaps in the knowledge graph.

5.2.4. Knowledge Graph Fusion

Many knowledge graphs contain links to other knowledge graphs. Those are often created automatically [53]. Interlinks between knowledge graphs can be used to fill gaps in one knowledge graph from information defined in another knowledge graph. If a mapping both on the instance and on the schema level is known, it can be exploited for filling gaps in knowledge graphs on both sides.

One work in this direction is presented by Bryl and Bizer [10], where different language versions of DBpedia (each of which is a knowledge graph of its own) are used to fill missing values in the English language DBpedia (the one which is usually meant when referring to *DBpedia*).

Dutta et al. [18] propose a probabilistic mapping between knowledge graphs. Based on distributions of types and properties, they create a mapping between knowledge graphs, which can then be used to derive additional, missing facts in the knowledge graphs. To that end, the type systems used by two knowledge graphs are mapped to one another. Then, types holding in one knowledge graph can be used to predict those that should hold in another.

6. Approaches for Error Detection in Knowledge Graphs

Like completion methods discussed in the previous section, methods for identifying errors in knowledge graphs can target various types of information, i.e., type assertions, relations between individuals, literal values, and knowledge graph interlinks.

In this section, we survey methods for detecting errors in knowledge graphs. Like for the previous section, we distinguish internal and external methods, and further group the approaches by the underlying methods used.

6.1. Internal Methods

Internal methods use only the information given in a knowledge graph to find out whether an axiom in the knowledge graph is plausible or not.

6.1.1. Classification

For the building Knowledge Vault, Dong et al. use classification to tell relations which should hold in a knowledge graph from those which should not [16]. Like the work by Muñoz et al. discussed above, each

relation is used as an instance in the classification problem, with the existence of the relation in the knowledge graph being used as a binary class. This classification is used as a cleansing step after the knowledge extraction process. While the creation of *positive* training examples from the knowledge graph is quite straight forward, the authors propose the creation of *negative* training examples by applying a *Local Closed World Assumption*, assuming that a relation r between two entities e_1 and e_2 does not hold if it is not present in the knowledge graph, and there is a relation r between e_1 and another e_3.

6.1.2. Reasoning

Reasoning is a technique from the semantic web community that, given a set of axioms, determines whether that set is free of contradictions or not [45]. To that end, a rich ontology is required, which defines the possible types of nodes and edges in a knowledge graph, as well as the restrictions that hold on them. For example, if a person is defined to be the capital of a state, this is a contradiction, since capitals are cities, and cities and persons are disjoint, i.e., no entity can be a city and a person at the same time.

Reasoning is often used at the building stage of a knowledge graph, i.e., when new axioms are about to be added. NELL and PROSPERA perform reasoning at that point to determine whether the new axiom is plausible or not, and discard implausible ones [11,52]. For real knowledge graphs, reasoning can be difficult due to the presence of errors and noise in the data [68, 34].

Works using reasoning as a refinement operation for knowledge graphs have also been proposed. However, many knowledge graphs, such as DBpedia, come with ontologies that are not rich enough to perform reasoning for inconsistency detection – for example, they lack class disjointness assertions needed for an inference as in the example above. Therefore, approaches exploiting reasoning are typically used in conjunction with methods for enriching ontologies, such as statistical methods, as proposed by Töpper et al. [78], association rule mining, as proposed by Lehmann and Bühmann [39], or inductive logic programming, as proposed by Ma et al. [46]. In all of those works, the ontology at hand is enriched with further axioms, which can then be used for detecting inconsistencies. For example, if a reasoner concludes that an entity should both be a person and an organization, and from the enrichment steps has a disjointness axiom between the two types added, a reasoner can state that one out of a few axioms in the knowledge graph has to be wrong.

In [66], a light-weight reasoning approach is proposed to compare actual and defined domains and ranges of relations in a knowledge graph schema. The authors propose a set of heuristics for fixing the schema if the actual and the defined domain or range strongly deviate.

6.1.3. Outlier Detection

Outlier detection or *anomaly detection* methods deal aim at identifying those instances in a dataset that deviate from the majority from the data, i.e., that follow different characteristics than the rest of the data [30,12].

As outlier detection in most cases deals with *numeric* data, numeric literals are a natural target for those methods. In [85], we have proposed the application of different univariate outlier detection methods (such as interquartile range or kernel density estimation) to DBpedia. While outlier detection does not necessarily identify errors, but also natural outliers (such as the population of very large cities), it has been shown that the vast majority of outliers identified are actual errors in DBpedia, mostly resulting from parsing errors.

To lower the influence of natural outliers, an extension of that approach has been presented in [19], where the instance set under inspection is first split into smaller subsets. For example, population values are inspected for countries, cities, and towns in isolation, thus, the distributions are more homogenous, which leads to a higher precision in error identification. Furthermore, the approach foresees *cross-checking* found outliers with other knowledge graphs in order to further reduce the influence of natural outliers, which makes it a mixed approach with both an internal and an external component.

In [61], we have shown that outlier detection is not only applicable to numerical values, but also to other targets, such as knowledge graph interlinks. To that end, the interlinks are represented as a multi-dimensional feature vector, e.g., with each type of the respective entity in both knowledge graphs being a binary feature. In that feature space, standard outlier detection techniques such as *Local Outlier Factor* [9] or *cluster-based outlier detection* [27] can be used to assign outlier scores. Based on those scores, *implausible* links, such as a `owl:sameAs` assertion between a person and a book, can be identified based only on the

overall distribution of all links, where such a combination is infrequent.

In [63], we have proposed a statistical method for finding wrong statements within a knowledge graph. Although not using established any standard outlier detection algorithm, the idea is quite similar. For each type of relation, like in the interlinking case above, we compute the characteristic distribution of subject and object types for each property. Edges in the graph whose subject and object type deviate from the characteristic distributions are then identified as potential errors.

6.1.4. Graph-based Methods

Knowledge graphs, by nature, form a graph. Hence, graph-based measures – like degree, clustering coefficient, centrality etc. – can be used to compute scores for nodes and edges in the knowledge graph. Guéret et al. [25] use such scores to define metrics for identifying wrong interlinks between knowledge graphs. There method compares the actual distributions of those metrics to the ones that are ideally expected – e.g., a power-law like distribution for the degree of entities [48] – and marks links that deviate from those expected distributions as suspicious.

6.2. External Methods

Purely external methods for error detection in knowledge graphs are still rare. One of the few works is *DeFacto* [40]. The system uses a database of lexicalizations for predicates in DBpedia. Based on those lexicalizations, it transforms statements in DBpedia to natural language sentences, and uses a web search engine to find web pages containing that sentence. Sentences with no or only very few web pages supporting the sentences are then assigned a low confidence score.

7. Findings from the Survey

From the survey in the last two sections, we can observe that there are quite a few works proposed for knowledge graph refinement, both for automatic completion and for error detection. Tables 2 to 4 sum up the results from the previous section.

By taking a closer look at those results, we can derive some interesting findings, both with respect to the approaches, as well as with respect to evaluation methodologies.

7.1. Approaches

A first interesting observation is that our distinguishing into completion and error detection is a strict one. That is, there exist no approaches which do *both* completion and correction at the same time. The only exception we found is the pairing of the two approaches *SDType* and *SDValidate* [63], which are two closely related algorithms which share the majority of the computations and can output both completion axioms and errors.

For many of the approaches, it is not obvious why whey were only used for one purpose. For example, many of the probabilistic and NLP-based completion approaches seek for evidence for missing relations, e.g., by means of scanning text corpora. In principle, they could also be used for error detection by flagging statements for which *no* evidence was found.

Furthermore, in particular in the machine learning area, approaches exist which can be used for simultaneously creating a predictive model and creating weights for pieces of information. For example, random forests can assign weights to attributes [43], whereas boosting assign weights to instances [21], which can also be interpreted as outlier scores [13]. Such approaches could be a starting point for developing methods for simultaneous completion and error detection in knowledge graphs.

Along the same lines, there are hardly any among the error detection approaches which are also suitable for *correcting* errors, i.e., suggest fixes for the errors found. Here, a combination between completion and error detection methods could be of great value: once an error is detected, the erroneous axiom(s) could be removed, and a correction algorithm could try to find a new (and, in the best case, more accurate) replacement for the removed axiom(s).

In addition to the strict separation of completion and correction, we also observe that most of the approaches focus on only one target, i.e., types, relations, literals, etc. Approaches that simultaneously try to complete or correct, e.g., type and relation assertions in a knowledge graph, are also quite rare.

For the approaches that perform completion, all works examined in this survey try to add missing types for or relations between *existing* entities in the knowledge graph. In contrast, we have not observed any approaches which populate the knowledge graph with *new* entities. Here, *entity set expansion* methods, which have been deeply investigated in the NLP field [59,72,83], would be an interesting fit to further in-

Table 2: Overview of knowledge graph completion approaches (part 1). Abbreviations used: Target (T=Types, R=Relations, S=Schema), Type (I=internal, E=external), Evaluation (EP=ex post, PG=partial gold standard, either available (a) or unavailable (n/a), KG=evaluation against knowledge graph, SV=split validation, CV=cross validation), Metrics (P/R=precision and recall, A=accuracy, AUC-PR=area under precision-recall-curve, T=total new statements). Comp.: evaluation or materialization carried out on whole knowledge graph or not, Performance: computational performance reported or not.

Paper	Target	Type	Methods and Sources	Knowledge Graph(s)	Eval.	Metrics	Whole	Comp.
Neville/Jenssen [54]	T	I	Bayesian Classification	US Securities Exchange Commission (excerpt)	KG (SV)	A	yes	no
Paulheim [60]	T	I	Association Rule Mining	DBpedia	EP	P, T	no	yes
Homoceanu et al. [32]	T	I	entropy based	BTC	PG (n/a)	P/R	no	no
Nickel et al. [56]	T	I	Matrix Factorization	YAGO	KG (CV)	P/R, AUC-PR	yes	yes
Paulheim/Bizer [62,63]	T	I	Likelihood based	DBpedia, OpenCyc, Nell	KG, EP	P/R, T	yes	no
Nuzzolese et al. [57]	T	E	different machine learning methods, Wikipedia link graph	DBpedia	PG (a)	P/R	yes	no
Gangemi et al. [23]	T	E	NLP on Wikipedia abstracts	DBpedia	PG (a)	P/R	no	yes
Kliegr [36]	T	E	NLP on Wikipedia abstracts	DBpedia	PG (n/a)	P/R	no	no
Aprosio et al. [3]	T	E	K-NN classification, different DBpedia language editions	DBpedia	PG (n/a)	P/R	yes	no
Dutta et al. [18]	T	E	Knowledge graph fusion (statistical)	NELL, DBpedia	PG (a)	P/R	no	no
Sleeman/Finin [75]	T	I, E	SVM, using other KGs	DBpedia, Freebase, Arnetminer	KG	P/R	no	no
Völker/Niepert [80]	S	I	Association Rule Mining	DBpedia, data.gov.uk	KG	P/R	yes	no
Fleischhacker/Völker [20]	S	I	Association Rule Mining	DBpedia	PG	P/R	yes	no
Töpper et al. [78]	S	I	Statistical Methods	DBpedia	KG	A,T	no	no

Table 3: Overview of knowledge graph completion approaches (part 2). Abbreviations used: Target (T=Types, R=Relations, S=Schema), Type (I=internal, E=external), Evaluation (EP=ex post, PG=partial gold standard, either available (a) or unavailable (n/a), KG=evaluation against knowledge graph, SV=split validation, CV=cross validation), Metrics (P/R=precision and recall, A=accuracy, AUC-PR=area under precision-recall-curve, T=total new statements). Comp.: evaluation or materialization carried out on whole knowledge graph or not, Performance: computational performance reported or not.

Paper	Target	Type	Methods and Sources	Knowledge Graph(s)	Eval.	Metrics	Whole	Comp.
Socher et al. [76]	R	I	Neural Tensor Network	WordNet, Freebase	KG (SV)	A	no	no
Oren et al. [58]	R	I	Co-occurence (likelihood)	different smaller ones	KG	P/R	no	yes
Galárraga et al. [22]	R	I	Association Rule Mining	YAGO, DBpedia	KG, EP	A,T	yes	yes
Bryl and Bizer [10]	R	E	Fusion of DBpedia language editions	DBpedia	PG (a)	A	no	no
Apriosio et al. [4]	R	E	Distant supervision and NLP techniques on Wikipedia text corpus	DBpedia	KG	P/R	no	no
Lange et al. [38]	R	E	Pattern learning on Wikipedia abstracts	(DBpedia)	KG (CV)	P/R	yes	yes
Wu et al. [86]	R	E	Pattern learning on Wikipedia articles, Web search	(DBpedia)	KG (CV)	P/R	no	no
Mintz et al. [50]	R	E	Distant supervision and NLP techniques on Wikipedia text corpus	Freebase	KG	P/R	no	no
West et al. [84]	R	E	search engines	Freebase	KG	P/R, rank	no	no
Mu noz et al. [51]	R	E	Statistical measures, machine learning, using Wikipedia tables	DBpedia	PG (a)	P/R	yes	no
Paulheim/Ponzetto [65]	T, R	E	Statistical measures, Wikipedia list pages	DBpedia	–	–	no	no

Table 4: Overview of error detection approaches. Abbreviations used: Target (T=Types, R=Relations, L=Literals, I=Interlinks, S=Schema), Type (I=internal, E=external), Evaluation (EP=ex post, PG=partial gold standard, either available (a) or unavailable (n/a), KG=evaluation against knowledge graph, SV=split validation, CV=cross validation), Metrics (P/R=precision and recall, A=accuracy, AUC=area under curve (ROC or precision-recall), T=total new statements, RMSE=Root Mean Squared Error). Comp.: evaluation or materialization carried out on whole knowledge graph or not, Performance: computational performance reported or not.

Paper	Target	Type	Methods and Sources	Knowledge Graph(s)	Eval.	Metrics	Whole	Comp.
Ma et al. [46]	T	I	Reasoning, Association Rule Mining	DBpedia, Zhishi.me	EP	P, T	yes	no
Dong et al. [16]	R	I	Classification	Knowledge Vault	KG (SV)	AUC-PR	yes	no
Nakashole et al. [52]	R	I	Reasoning	Prospera	EP	P, T	yes	yes
Paulheim/Bizer [63]	R	I	Probabilistic	DBpedia, NELL	EP	P	yes	no
Lehmann et al. [40]	R	E	Text pattern induction, Web search engines	DBpedia	KG (CV)	P/R, AUC-ROC, RMSE	no	yes
Töpper et al. [78]	R,S	I	Statistical methods, Reasoning	DBpedia	EP	P, T	yes	no
Lehmann/Bühmann [39]	R,T	I	Reasoning, ILP	DBpedia, Open Cyc, seven smaller ontologies	EP	A	yes	yes
Wienand/Paulheim [85]	L	I	Outlier Detection	DBpedia	EP	P, T	no	no
Fleischhacker et al. [19]	L	I, E	Outlier Detection and Data Fusion with other KG	DBpedia, NELL	EP	AUC-ROC	no	no
Paulheim [61]	I	I	Outlier Detection	DBpedia + two linked graphs	PG (a)	P/R, AUC-ROC	yes	no
Gueret et al. [25]	I	I	Network measures	Geonames and others	PG (a)	P/R	no	no
Péron et al. [66]	S	I	Statistical methods Reasoning	DBpedia	EP	–	yes	yes

crease the coverage of knowledge graphs, especially for less well-known long tail entities.

Another interesting observation is that, although the discussed works address knowledge *graphs*, only very few of them are, in the end, genuinely graph-based approaches. In many cases, simplistic transformations to a propositional problem formulation are taken. Here, methods from the graph mining literature still seek their application to knowledge graphs. In particular, for many of the methods applied in the works discussed above – such as outlier detection or association rule mining – graph-based variants have been proposed in the literature [2,35]. Likewise, graph kernel functions – which can be used in Support Vector Machines as well as other machine learning algorithms – have been proposed for RDF graphs [33,44,15] and hence could be applied to many web knowledge graphs.

7.2. Evaluation Methodologies

For evaluation methodologies, our first observation is that there are various different evaluation metrics being used in the papers examined. There is a clear tendency towards precision and recall (or precision and total number of statements for ex post evaluations) are the most used metrics, with others – such as ROC curves, accuracy, or Root Mean Squared Error – occasionally being used as well.

With respect to the overall methodology, the results are more mixed. Evaluations using the knowledge graph as a silver standard, ex post evaluations, and evaluations based on partial gold standards appear at equal frequency, with ex post validations mostly used for error detection. The latter is not too surprising, since due to the high quality of most knowledge graphs used for the evaluations, partial gold standards based on random samples are likely to contain only few errors. For partial gold standards, it is crucial to point out that the majority of authors make those partial gold standards public[13], which allows for replication and comparison.

The major knowledge graph used in the evaluations is DBpedia. This, in principle, makes the results comparable to a certain extent, although roughly each year, a new version of DBpedia is published, so that papers from different years are likely to be evaluated on slightly different knowledge graphs.

That being said, we have observed that roughly two out of three approaches evaluated on DBpedia are *only* evaluated on DBpedia. Along the same lines, about half of the approaches reviewed in this survey are only evaluated on *one* knowledge graph. This, in many cases, limits the significance of the results. For some works, it is clear that they can only work on a specific knowledge graph, e.g., DBpedia, *by design*, e.g., since they exploit the implicit linkage between a DBpedia entity and the corresponding Wikipedia page.

As discussed in section 2, knowledge graphs differ heavily in their characteristisc. Thus, for an approach evaluated on only one graph, it is unclear whether it would perform similarly on another knowledge graph with different characteristics, or whether it exploits some (maybe not even obvious) characteristics of that knowledge graph, and/or overfits to particular characteristics of that graph.

Last, but not least, we have observed that only a minority of approaches have been evaluated on a whole, large-scale knowledge graph. Moreover, statements about computational performance are only rarely included in the corresponding papers[14]. In the age of large-scale knowledge graphs, we think that this is a dimension that should not be neglected.

In order to make future works on knowledge graph evolution comparable, it would be useful to have a common selection of benchmarks. This has been done in other fields of the semantic web as well, such as for schema and instance matching [17], reasoning [6], or question answering [14]. Such benchmarks could serve both for comparison in the qualitative as well as the computational performance.

8. Conclusion

In this paper, we have presented a survey on knowledge base refinement methods. We distinguish completion from error detection, and internal from external methods. We have shown that a larger body of works exist which apply different methods, ranging from techniques from the machine learning field to NLP related techniques.

[13] For this survey, we counted a partial gold standard as public if there was a working download link in the paper, but we did not make any additional efforts to search for the gold standard, such as contacting the authors.

[14] Even though we were relaxed on this policy and counted also informal statements about the computational performance as a performance evaluation.

The survey has revealed that there are, at the moment, rarely any approaches which simultaneously try to improve completeness and correctness of knowledge graphs, and usually only address one target, such as type or relation assertions, or literal values. Holistic solutions which simultaneously improve the quality of knowledge graphs in many different aspects are currently not observed.

Looking at the evaluation methods, the picture is quite diverse. Different methods are applied, using either the knowledge graph itself as silver standard, using a partial gold standard, or performing an ex post evaluation, are about equally distributed. Furthermore, approaches are often only evaluated on one specific knowledge graph. This makes it hard to compare approaches and make general statements on their relative performance.

In addition, scalability issues are only rarely addressed by current research works. In the light of the advent of web-scale knowledge graphs, however, this is an aspect which will be of growing importance.

To sum up, this survey shows that automatic knowledge graph refinement is a relevant and flowering research area. At the same time, this survey has pointed out some uncharted territories on the research map, which we hope will inspire researchers in the area.

References

[1] Maribel Acosta, Amrapali Zaveri, Elena Simperl, Dimitris Kontokostas, Sören Auer, and Jens Lehmann. Crowdsourcing linked data quality assessment. In *The Semantic Web–ISWC 2013*, pages 260–276. Springer, 2013.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, pages 1–63, 2014.

[3] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of dbpedia exploiting wikipedia cross-language information. In *The Semantic Web: Semantics and Big Data*, pages 397–411. Springer, 2013.

[4] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Extending the coverage of dbpedia properties using distant supervision over wikipedia. In *NLP&DBpedia*, 2013.

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International journal on semantic web and information systems*, 5(3):1–22, 2009.

[6] Jürgen Bock, Peter Haase, Qiu Ji, and Raphael Volz. Benchmarking owl reasoners. In *Proc. of the ARea2008 Workshop, Tenerife, Spain (June 2008)*, 2008.

[7] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[8] Antoine Bordes and Evgeniy Gabrilovich. Constructing and mining web-scale knowledge graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1967–1967, 2014.

[9] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. 29(2):93–104, 2000.

[10] Volha Bryl and Christian Bizer. Learning conflict resolution strategies for cross-language wikipedia data fusion. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 1129–1134, 2014.

[11] Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM, 2010.

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 2009.

[13] Nathalie Cheze and Jean-Michel Poggi. Iterated boosting for outlier detection. In *Data Science and Classification*, pages 213–220. Springer, 2006.

[14] Philipp Cimiano, Vanessa Lopez, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. Multilingual question answering over linked data (qald-3): Lab overview. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 321–332. Springer, 2013.

[15] Gerben KD de Vries. A fast approximation of the weisfeiler-lehman graph kernel for rdf data. In *Machine Learning and Knowledge Discovery in Databases*, pages 606–621. 2013.

[16] Xin Luna Dong, K Murphy, E Gabrilovich, G Heitz, W Horn, N Lao, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion, 2014.

[17] Zlatan Dragisic, Kai Eckert, Jerome Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jimenez-Ruiz, Andreas Kempf, Patrick Lambrix, et al. Results of theontology alignment evaluation initiative 2014. In *International Workshop on Ontology Matching*, pages 61–104, 2014.

[18] Arnab Dutta, Christian Meilicke, and Simone Paolo Ponzetto. A probabilistic approach for integrating heterogeneous knowledge sources. In *The Semantic Web: Trends and Challenges*, pages 286–301. Springer, 2014.

[19] Daniel Fleischhacker, Heiko Paulheim, Volha Bryl, Johanna Völker, and Christian Bizer. Detecting errors in numerical linked data using cross-checked outlier detection. In *The Semantic Web–ISWC 2014*, pages 357–372. Springer, 2014.

[20] Daniel Fleischhacker and Johanna Völker. Inductive learning of disjointness axioms. In *On the Move to Meaningful Internet Systems: OTM 2011*, pages 680–697. Springer, 2011.

[21] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[22] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*,

pages 413–422, 2013.

[23] Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[24] Lise Getoor and Christopher P Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.

[25] Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *The Semantic Web: Research and Applications*, pages 87–102. Springer, 2012.

[26] Harry Halpin, PatrickJ. Hayes, JamesP. McCusker, DeborahL. McGuinness, and HenryS. Thompson. When owl:sameAs IsnâĂŹt the Same: An Analysis of Identity in Linked Data. In *The Semantic Web âĂŞ ISWC 2010*, pages 305–320. Springer Berlin Heidelberg, 2010.

[27] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641–1650, 2003.

[28] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

[29] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule miningâĂŤa general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64, 2000.

[30] Victoria J Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[31] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792, 2011.

[32] Silviu Homoceanu, Philipp Wille, and Wolf-Tilo Balke. Proswip: Property-based data access for semantic web interactive programming. In *The Semantic Web–ISWC 2013*, pages 184–199. Springer, 2013.

[33] Yi Huang, Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A scalable kernel approach to learning in semantic graphs with applications to linked data. In *1st Workshop on Mining the Future Internet*, 2010.

[34] Qiu Ji, Zhiqiang Gao, and Zhisheng Huang. Reasoning with noisy semantic data. In *The Semanic Web: Research and Applications*, pages 497–502. Springer, 2011.

[35] Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(01):75–105, 2013.

[36] TomÃąÅą Kliegr. Linked hypernyms: Enriching dbpedia with targeted hypernym discovery. *Web Semantics: Science, Services and Agents on the World Wide Web*.

[37] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning*, pages 282–289, 2001.

[38] Dustin Lange, Christoph Böhm, and Felix Naumann. Extracting structured information from wikipedia articles to populate infoboxes. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1661–1664, Toronto, Canada, 0 2010.

[39] Jens Lehmann and Lorenz Bühmann. Ore-a tool for repairing and enriching knowledge bases. In *The Semantic Web–ISWC 2010*, pages 177–193. Springer, 2010.

[40] Jens Lehmann, Daniel Gerber, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. Defacto-deep fact validation. In *The Semantic Web–ISWC 2012*, pages 312–327. Springer, 2012.

[41] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, SÃűren Auer, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2013.

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

[43] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[44] Uta Lösch, Stephan Bloehdorn, and Achim Rettinger. Graph kernels for rdf data. In *The Semantic Web: Research and Applications*, pages 134–148. 2012.

[45] Marko Luther, Thorsten Liebig, Sebastian Böhm, and Olaf Noppens. Who the heck is the father of bob? In *The Semantic Web: Research and Applications*, pages 66–80. Springer, 2009.

[46] Yanfang Ma, Huan Gao, Tianxing Wu, and Guilin Qi. Learning disjointness axioms with association rule mining and its application to inconsistency detection of linked data. In Dongyan Zhao, Jianfeng Du, Haofen Wang, Peng Wang, Donghong Ji, and Jeff Z. Pan, editors, *The Semantic Web and Web Science*, volume 480 of *Communications in Computer and Information Science*, pages 29–41. Springer, 2014.

[47] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Conference on Innovative Data Systems Research*, 2015.

[48] Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. Graph structure in the web—revisited: a trick of the heavy tail. In *Proceedings of the companion publication of the 23rd international conference on World wide web*, pages 427–432, 2014.

[49] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[50] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[51] Emir Muñoz, Aidan Hogan, and Alessandra Mileo. Triplifying wikipedia's tables. In *Linked Data for Information Extraction*, 2013.

[52] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 227–236. ACM, 2011.

[53] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semantic Web Journal*, 2015.

[54] Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[55] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*, 2015.

[56] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 271–280, 2012.

[57] Andrea Giovanni Nuzzolese, Aldo Gangemi, Valentina Presutti, and Paolo Ciancarini. Type inference through the analysis of wikipedia links. In *LDOW*, 2012.

[58] Eyal Oren, Sebastian Gerke, and Stefan Decker. Simple algorithms for predicate suggestions using similarity and co-occurrence. In *The Semantic Web: Research and Applications*, pages 160–174. Springer, 2007.

[59] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 938–947, 2009.

[60] Heiko Paulheim. Browsing linked open data with auto complete. *Semantic Web Challenge*, 2012.

[61] Heiko Paulheim. Identifying wrong links between datasets by multi-dimensional outlier detection. In *International Workshop on Debugging Ontologies and Ontology Mappings*, 2014.

[62] Heiko Paulheim and Christian Bizer. Type inference on noisy rdf data. In *The Semantic Web–ISWC 2013*, pages 510–525. Springer, 2013.

[63] Heiko Paulheim and Christian Bizer. Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):63–86, 2014.

[64] Heiko Paulheim and Johannes Fürnkranz. Unsupervised generation of data mining features from linked open data. In *2nd international conference on web intelligence, mining and semantics*, page 31. ACM, 2012.

[65] Heiko Paulheim and Simone Paolo Ponzetto. Extending dbpedia with wikipedia list pages. In *1st International Workshop on NLP and DBpedia*, 2013.

[66] Youen Péron, Frédéric Raimbault, Gildas Ménier, and Pierre-François Marteau. On the detection of inconsistencies in RDF data sets and their correction at ontological level. Technical report, 2011.

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

[68] Axel Polleres, Aidan Hogan, Andreas Harth, and Stefan Decker. Can we ever catch up with the web? *Semantic Web*, 1(1):45–52, 2010.

[69] Petar Ristoski and Heiko Paulheim. A comparison of propositionalization strategies for creating features from linked open data. In *Linked Data for Knowledge Discovery*, 2014.

[70] Giuseppe Rizzo and Raphaël Troncy. NERD: evaluating named entity recognition tools in the web of data. In *Workshop on Web Scale Knowledge Extraction (WEKEX'11)*, 2011.

[71] Samuel Sarjant, Catherine Legg, Michael Robinson, and Olena Medelyan. All you can eat ontology-building: Feeding wikipedia to cyc. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 341–348. IEEE Computer Society, 2009.

[72] Luis Sarmento, Valentin Jijkuon, Maarten de Rijke, and Eugenio Oliveira. More like these: growing entity classes from seeds. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 959–962. ACM, 2007.

[73] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In *International Semantic Web Conference*, 2014.

[74] Carlos N Silla Jr and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[75] Jennifer Sleeman and Tim Finin. Type prediction for efficient coreference resolution in heterogeneous semantic graphs. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 78–85. IEEE, 2013.

[76] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 926–934. Curran Associates, Inc., 2013.

[77] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *16th international conference on World Wide Web*, pages 697–706, 2007.

[78] Gerald Töpper, Magnus Knuth, and Harald Sack. Dbpedia ontology enrichment for inconsistency detection. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 33–40. ACM, 2012.

[79] G Tsoumakas et al. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[80] Johanna Völker and Mathias Niepert. Statistical schema induction. In *The Semantic Web: Research and Applications*, pages 124–138. Springer, 2011.

[81] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[82] Jörg Waitelonis, Nadine Ludwig, Magnus Knuth, and Harald Sack. Whoknows? evaluating linked data heuristics with a quiz that cleans up dbpedia. *Interactive Technology and Smart Education*, 8(4):236–248, 2011.

[83] Richard C Wang and William W Cohen. Iterative set expansion of named entities using the web. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 1091–1096. IEEE, 2008.

[84] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*, pages 515–526, 2014.

[85] Dominik Wienand and Heiko Paulheim. Detecting incorrect numerical data in dbpedia. In *The Semantic Web: Trends and Challenges*, pages 504–518. Springer, 2014.

[86] Fei Wu, Raphael Hoffmann, and Daniel S Weld. Information extraction from wikipedia: Moving down the long tail. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–739. ACM, 2008.

[87] Amrapali Zaveri, Dimitris Kontokostas, Mohamed A Sherif, Lorenz Bühmann, Mohamed Morsey, Sören Auer, and Jens

Lehmann. User-driven quality evaluation of dbpedia. In *9th International Conference on Semantic Systems (I-SEMANTICS '13)*, 2013.

[88] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, Sören Auer, and Pascal Hitzler. Quality assessment methodologies for linked open data. *Submitted to Semantic Web Journal*, 2013.

[89] Antoine Zimmermann, Christophe Gravier, Julien Subercaze, and Quentin Cruzille. Nell2rdf: Read the web, and turn it into rdf. In *Knowledge Discovery and Data Mining meets Linked Open Data*, pages 2–8, 2013.