Knowledge Graph Quality Management: A Comprehensive Survey

Bingcong Xue and Lei Zou

Abstract—As a powerful expression of human knowledge in a structural form, knowledge graph (KG) has drawn great attention from both the academia and the industry and a large number of construction and application technologies have been proposed. Large-scale knowledge graphs such as DBpedia, YAGO and Wikidata are published and widely used in various tasks. However, most of them are far from perfect and have many quality issues. For example, they may contain inaccurate or outdated entries and do not cover enough facts, which limits their credibility and further utility. Data quality has a long research history in the field of traditional relational data and recently attracts more knowledge graph experts. In this paper, we provide a systematic and comprehensive review of the quality management on knowledge graphs, covering overall research topics about not only quality issues, dimentions and metrics, but also quality management processes from quality assessment and error detection, to error correction and KG completion. We categorize existing works in terms of target goals and used methods for better understanding. In the end, we discuss some key issues and possible directions on knowledge graph quality management for further research.

Index Terms—Knowledge graph, quality management, evaluation, error detection, error correction, completion

1 Introduction

Recent years have witnessed vigorous development of knowledge graph (KG) construction and application. KG expresses real-world entities and relationships in a structural way and has great potential to carry human knowledge and promote the development of artificial intelligence. Many large-scale knowledge graphs, such as DBpeida [1], YAGO [2], Wikidata [3], NELL [4] and KnowledgeVault [5], are constructed from various structured, semi-structured or unstructured data sources. They have been widely used in several real-world applications, from information retrieval [6], question answering [7], [8], to recommender systems [9], [10] and domain-specific tasks [11], [12].

However, as these graphs are often extracted and fused from different sources automatically or semi-automatically, they are far from perfect and have a large variation in data quality [13]. For example, errors and conflicts may come from the data sources or the extraction and fusion stages, and the KGs can hardly cover all the facts we need so that incomplete problem exists. Quality issues have big impact on the credibility and usability of the knowledge graphs. In order to further increase the utility of such knowledge graphs in downstream tasks, quality management processes need to be taken into consideration carefully, from quality assessment, problem discovery (e.g., error and inconsistency detection) to quality improvement (e.g., error correction and graph completion).

Research on data quality has a long history and it can be traced back to 1990s, when the MIT Total Data Quality Management (TDQM) program was formally established to treat data quality as a specialized research field [14]. Since then, a considerable amount of literature has been published on data quality dimensions and metrics [15], [16], [17]. And a large number of methods and tools are developed for assessment, detection, and repair of data quality problems [18], [19], [20]. With the advent of the Big Data Era, the characteristics of the 4 V's (Volume, Velocity, Variety, Value) bring new challenges to quality management [21], [22], [23]. More and more researchers turn eyes to newly emerging data structures, such as the widely-used knowledge graphs, and many graph-specific quality management methods are proposed.

Quality management methods on traditional relational data are difficult to be applied to KGs directly for at least four reasons. First, unlike relational data, graphs are semi-structured and often do not come with a schema to specify the integrity and semantics of the data. Heterogeneity and flexibility make the structures more complex. Second, the Semantic Web and knowledge graphs typically follow the Open World Assumption (OWA) [24], where a statement not included in the KG can be wrong or just absent. So it's difficult to distinguish wrong tuples from missing ones. What's more, real-world KGs often contain massive noise, and the assumption widely adopted in traditional technologies that the data is basically correct may not hold. Last but not least, due to the scale of the real-life graphs which is typically beyond the capacity of existing methods, a direct application of such techniques often suffers an unbearable

- Bingcong Xue is with the Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China. E-mail: xuebingcong@pku.edu.cn.
- Lei Zou is with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China. E-mail: zoulei@pku.edu.cn.

Manuscript received 25 September 2021; revised 15 December 2021; accepted 29 January 2022. Date of publication 10 February 2022; date of current version 3 April 2023.
This work was supported in part by the National Key R&D Program of China under Grants 2020AAA0105200 and in part by NSFC under Grants 61932001, 61961130390, and U20A20174.
(Corresponding author: Lei Zou.)
Recommended for acceptance by Z. Wang.
Digital Object Identifier no. 10.1109/TKDE.2022.3150080

time and complexity. Therefore, new solutions for knowledge graphs are in urgent need and gradually developed.

In this paper, we aim to review recent researches on the knowledge graph quality management, from theory to practice, and provide a systematic and comprehensive survey on related works, hoping to give an intuitive and clear overview and inspire new opinions and methods to readers. We notice that many related but not identical surveys on knowledge graphs and data quality have been published in recent years. Some focus on features and construction technologies of existing knowledge graphs [25], [26], [27], some explore reasoning technologies on knowledge graphs as well as their applications [28], [29], while others focus on statistical relation learning and embedding learning methods [30], [31], [32]. On the other hand, traditional data quality research is still advancing, with surveys about data quality metrics and assesment [20], [33], data quality theory [34], [35] and tools [18], [19], and challenges in Big Data Era [22], [36]. There are also some works studying knowledge graph quality by combining these two research fields together. For example, [16] reviews approaches for assessing the quality of Linking Open Data (LOD) and provides a comprehensive list of quality dimensions and metrics. [13], [37] and [38] experimentally evaluate the quality of existing knowledge graphs by some proposed metrics. And methodological researches for specific quality issues such as completeness [39] and duplication [40] are published. However, these studies remain narrow in focus dealing only with part of the quality issues and to date there are few studies that try to give a panoramic overview of knowledge graph quality management technologies, which covers not only definition of quality dimensions and metrics, but also the whole process from assessment, detection to improvement of knowledge graph quality problems.

The most related work to our focus is [41], written by Heiko Paulheim and published in 2017. In this paper, Paulheim surveys knowledge graph refinement technologies in terms of approaches and evaluation methods. He distinguishes KG completion from error detection, and internal from external methods, and further categorizes the methods by the refinement target such as entity types, relations and literal values. This is a good survey for knowledge graph quality management and has inspired a number of subsequent researches. At the same time, however, its taxonomy is mainly based on shallow features like internal or external resources and target type, failing to look deep into the methods used by different works. It focuses only on tasks of completion and error detection, and does not take into account other quality management aspects like quality assessment and error correction. Besides, various quality dimensions and metrics are not included in, leading to the fact that it pays too much attention to correctness and completeness and overlooks other issues such as timeliness and redundancy. And it doesn't contain latest methods published in recent years.

Out of the above reasons, we carrry out a deep and careful review of works on knowledge graph quality management, expecially those published in recent six years, and provide a comprehensive overview with in-depth analysis. Our main contributions are summarized as follows:

1) *Comprehensive and Newest Review.* We present a systematic and comprehensive review on all aspects of knowledge graph quality management, from theory to practice, including not only quality issues, dimensions and metrics, but also the whole quality management process from quality assessment and error detection, to error correction and KG completion.
2) *In-depth Taxonomies.* We categorize existing works on three orthogonal dimensions. For methods used, they are generally categorised into human-based, statistics/learning-based, rule-based, and hybrid approaches; for processing goals, they fall under three headings: (1) quality assessment, (2) problem discovery, and (3) quality improvement; for target dimensions, accuracy, consistency, completeness, timeliness and redundancy are adopted for classfication. This multi-dimensional taxonomy helps to better understand and analyse existing methods, which we believe will inspire more fancy ideas and technologies.
3) *Discussions and Outlook on Future Directions.* In the end of the article, we take a closer reflection and summary of the proposed methods, showing some interesting findings as well as providing several potential research directions.

The rest of the paper is organized in the following way. Section 2 gives a brief introduction on knowledge graph and data quality foundations, as well as our research objects and categorization. In Sections 3, 4, and 5, we present knowledge graph quality management technologies on human-based, statistics-based, and rule-based methods respectively. Then we introduce some hybrid approaches with more than one measure of human, rule and statistics in Section 6. Section 7 gives an in-depth discussion on the listed methods and presents some interesting findings and further directions. And in Section 8 we conclude the paper.

2 Preliminaries

2.1 Knowledge Graph and RDF Model

Knowledge base (KB) is a set of rules, facts and assumptions that stores knowledge in a machine understandable format [27]. The term *knowledge graph*, is first proposed by Google in 2012[1], which can be seen as a specification of KB that stores knowledge in the from of graphs. Following the definition of Ji *et al.* in [42], a knowledge graph is a multi-relational graph composed of entities and relations which are regarded as nodes and different types of edges respectively.

The W3C's Resource Description Framework (RDF)[2] is a general data model for knowledge representation. In RDF standard, each fact is represented in the form of (*subject, predicate, object*) (SPO) triples, where *subject* and *object* are entities and *predicate* reflects the relation between them. There are also predicates whose objects are literal values instead of entities, which are used to describe different attributes of the entities. For example, the statement "*the Capital of China is Beijing*" can be represented as (*sub:China,*

1. https://blog.google/products/search/introducing-knowledge-graph-things-not/
2. https://www.w3.org/RDF/

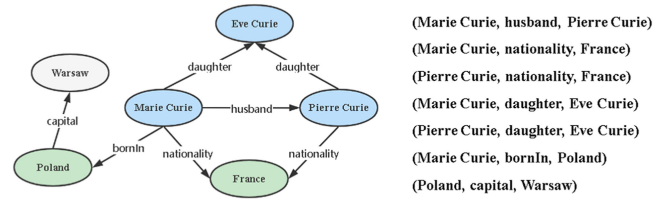

Fig. 1. An example of a knowledge graph and its corresponding RDF triples.

pred:capital, *obj:Beijing*) in RDF. Each triple is an atomic element and can be changed into nodes and edges to form into a graph in their visual presentation, called RDF graph.

Fig. 1 shows an example of a knowledge graph with its corresponding RDF triples. In this article, we will follow the RDF standard to represent knowledge graphs.

2.2 Data Quality Foundations

Data quality (DQ) has been studied for a long time and many definitions and assessment measures are proposed. However, there is still no uniform standard accepted by both academia and industry. One thing for sure is that data quality depends not only on its own characteristics, but also on the business environment being used, including business processes and business users [22].

2.2.1 Data Quality Definition

One generally accepted definition for data quality is ”fit for use” [43], [44], which means that the assessment of data quality is highly subject and context-dependent. It is not an absolute measure, only used to understand the suitability with specific applications, but is not sufficient to develop evaluation and improvement algorithms. In terms of machine implementation, a more technical explanation is ”free of defects” [45], by which algorithms can detect violation and errors based on a given criteria, in the form of logic rules or statistical thresholds, etc.

2.2.2 Data Quality Dimensions

Data quality dimensions give a way to assess data quality from different aspects, each of which is associated with various metrics and indicators to be calculated. What dimensions to choose depends on the data comsumer and downstream task. Generally, it can be divided into four categories: intrinsic, contextual, accessibility and representation [23], [46], as illustrated in Fig. 2. More essentially, these dimensions fall into two classes of intrinsic and extrinsic [47], where the former rely on the data itself and the latter are application-dependent. Intristic dimensions mainly include:

- Accuracy: It measures whether the data reflects the facts correctly, i.e., it is the degree to which the data is close to the realistic value.
- Consistency: It means that the data agrees with each other and is free of conflicts with respect to particular integrity constraints.
- Completeness: It describes whether the dataset contains all relevant data of interest, including levels of schema, property, types, etc. About completeness, there are Closed World Assumption (CWA), Open World Assumption (OWA) and Partial-Completeness Assumption (PCA) [48] to interpret the non-existent triples.
- Timeliness: It reflects the degree to which the data is up-to-date [49], and is useful in datasets that often change dynamically.
- Redundancy: It means that the dataset does not contain two identical objects (like entities or attributes) with different names.

It is important to keep in mind that these dimensions are not independent of each other and have complex inter-relationships. For example, a knowledge graph committed to covering more triples is quite likely to have a lower accuracy. Thus there is a trade-off among different quality issues for dataset constructors and they vary in different fields and application tasks. For data quality practitioners, selecting quality dimensions also depends on their specific needs.

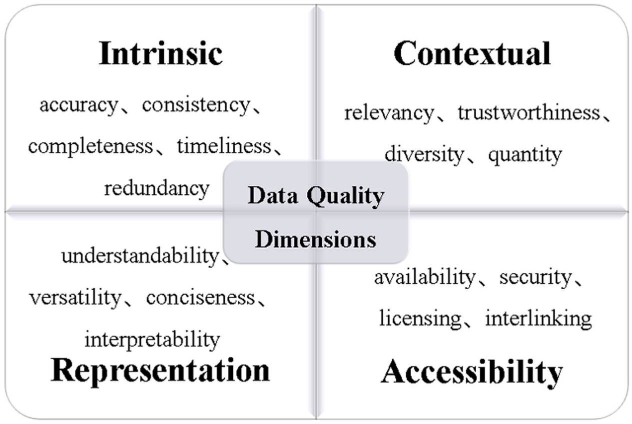

Fig. 2. Data quality dimensions and main characteristics.

2.2.3 Data Quality Metrics

As data quality dimensions are just abstract concepts, it is required to define specific metrics to apply and measure these dimensions in practice. These metrics need to build a connection with the underlying data in spite of intrinsic or extrinsic aspects. For example, accuracy can be defined as the percentage of the correct facts. Though the intrinsic metrics generally can be implemented without relying on external environment, this is not always the case. An instance is that the measure of completeness largely depends on its context.

Intrinsic and extrinsic metrics have interactions. As Sadiq *et al.* conclude in [50], the aim of intrinsic metrics is eventually contribute to an extrinsic metric, and the extrinsic metrics have to be tied to underlying intrinsic metrics. They have an overlapping relationship and both rely on downstream applications.

2.2.4 Data Lifecycle

There are a lifecycle and several processing transformations for data from its generation to actural applications, and quality issues can occur at any stage. This means that data quality consideration should be rooted along the whole pipeline. Fig. 3 shows a data lifecycle pipeline [23], [51]

Fig. 3. Data lifecycle from generation to application.

containing five steps, namely, data generation, information extraction, data integration, analysis and application.

2.2.5 Data Quality Management

In a previous survey on data quality measurement and monitoring tools [18], the authors divide data quality methodologies into four activities: data profiling, data quality measurement, data cleansing and continuous quality monitoring. We reconsider this question and classify data quality management into the following three processes:

1) *Quality Assessment.* This is the process of quality measurement and evaluation using pre-defined dimensions and metrics to check whether the data quality can meet the requirement of the application.
2) *Problem Discovery.* Different from assessment that only aiming at an overall perception of specific quality issues, problem discovery devotes to finding out the inherent wrong assertions, and further, to deriving higher level patterns to explain why these errors occur. The identified error assertions can be removed or modified, and the error patterns help to reveal the underlying causes, both of which contribute to the further quality improvement.
3) *Quality Improvement.* This is to improve the overall quality of the dataset. Considering the various quality dimensions, it can be calssified into two types: error correction, which involves various issues like inaccuracy, inconsistency and outdated, and completion, which improves the coverage of the dataset.

These three processes have a progressive relation and can be implemented at all stages of the data lifecycle.

2.3 Knowledge Graph Quality

As a specfic data type, researches on knowledge graph are in the same line with general data type. The definition, dimensions and metrics on data quality can be transferred to knowledge graphs, as those did in [16], [46]. Due to the particularity of the knowledge graph structure, there are also possibilities to develop new dedicated dimensions and metrics. In this paper, we focus on the intrisic features and choose five widely used indicators: accuracy, consistency, completeness, timeliness and redundancy, which are explained in Section 2.2.2.

Like the data lifecycle depicted in Fig. 3, knowledge graph also has a construction and application pipeline. That is, data sources' acquisition and evaluation before construction, knowledge extraction and fusion under construction, and interesting applications after construction [52]. Similarly, quality issues can happen and be processed at all of the stages.

Owing to the schemalessness, heterogeneity, Open World Assumpion, massive noise and scalability issues, directly applying traditional quality management methods on knowledge graphs faces some challenges and problems, which calls for new and dedicated solutions. At the same time, however, the structure and path characteristics of graphs bring extra opportunities and possibilities for the problem. Data quality and knowledge graph quality are by no means isolated and they can promote each other. Generic methods can be modified to fit for graphs and the development of tailored methods will promote data quality researches as well.

2.4 Coverage of This Article

Due to the wide range of the concepts, it is necessary to explain the focus and coverage of the article here. In this paper, we seek to review researches on knowledge graph quality management. That is to say, for target objects, we focus on knowledge graphs rather than generic data types. We pay attention to the whole process of quality management, from assessment, problem discovery, to quality improvement. In terms of the lifecycle of knowledge graphs, our focus is on the assessment and refinement of the constructed graphs, which omits the source evaluation before construction and is distinguished from the technologies of knowledge graph construction like extraction and fusion. For quality dimensions, we focus on the five intrisic features and bypass those application-dependent extrinsic ones.

Positioning at the given knowledge graphs and the intrisic dimensions contributes to focus on generic methods and technologies. It separates quality issues from different downstream tasks and is independent of different construction methods, which helps to better understand the essential thoughts of various quality management measures. Besides, since a large amount of knowledge graphs have been constructed and released, focusing on the after-construction technologies has more space for implement and evaluation, and frees the quality practitioners from the tedious construction processes.

Based on these ideas, we conduct a systematic review procedure by using inclusion and exclusion criteria to search and restrict related publications as [28] and [39] do. The search strategy is divided into three steps:

- Search on Google Scholar and get the first 100 results with keywords "knowledge graph quality". Check the publication lists of major data management and Semantic Web conferences including SIGMOD, ICDE, VLDB, WWW, ISWC and ESWC from 2016 to 2021 (note that the survey of Paulheim's [41] is up to 2015.), with at least one of the keywords ("quality", "knowledge graph", "knowledge base", "linked open data", "assessment", "validation", "refinement", "link prediction", "completion", "detect", "clean", "repair") appearing in the titles.
- Remove those not related to our purpose from the candidate publications by checking the titles, abstracts and sometimes the full articles.
- Search more relevant references and citations from those significant articles iteratively until no more key articles are found.

This procedure results in more than 1,000 candidate publications, and by iterative expansion and careful examination, the core articles are basically included, especially those published in recent years.

2.5 Taxonomies of This Article

Related works and methods are organized and categorized from three orthogonal aspects: technologies used, processing goals and target dimensions, which are explained next.

2.5.1 Technologies Used

Based on the inherent technologies used by different methods, they are divided into several groups:

(a) Human-based, where manpower plays an important role, of either experts or crowdsourcing.
(b) Statistics&Learning-based. This branch contains works of both traditional statistical methods like outlier detection and classic machine learning algorithms, as well as the embedding-based representation learning and neural networks.
(c) Rule-based, where rules of different forms are defined, extracted, checked and applied for quality management.
(d) Hybrid approaches, where techniques of human intelligence, statistical means and rule reasoning are combined in some way.

2.5.2 Processing Goals

As the data quality management process introduced in Section 2.2.5, methods are classified into three processing goals of quality assessment, problem discovery and quality improvement, where problem discovery can be further subdivided into false assertion recognition and error pattern derivation, and quality improvement contains error correction and graph completion.

2.5.3 Target Dimensions

We focus on intrinsic quality dimensions and articles can be grouped and classified according to their attention on various dimensions, i.e., accuracy, consistency, completeness, timeliness, and redundancy, as explained in Section 2.2.2.

3 Methods Based on Human

For problems that are diffcult to be solved by machines, manual methods are generally considered to be intuitive and credible, though sometimes cost and scalability issues exist. In this section we talk about methods based mainly on human intelligence, where the human can be both domain experts and crowdsourcing workers without specific skills, and they can participate in all the processes of quality management. Methods where manpower is introduced as extra external sources or combined with other technologies are left in the next sections.

3.1 Methods for Quality Assessment

Quality assessment of knowledge graphs is indispensable for downstream applications and subsequent improvement. The overall quality of KGs can quantify the fitness to various tasks, and fine-grained quality measurements on single predicates or classes give a way to feedback to the construction process and do further correction. But this problem has long been overlooked by academic research.

Manual evaluation is typically the main method to conduct quality assessment. Due to the scale of real-life KGs, it is not possible to exhaust all tuples, thus an alternative way is to evaluate on a sample set and the sample result is used to estimate the whole. The simplest and most widely used sampling technique is simple random sampling [13], but determining an appropriate number of samples is not easy: small sample sets prone to deviate from the real value and oversampling brings more labeling cost. To handle this, Ojha and Talukdar [53] build a novel crowdsourcing system, KGEval, for knowledge graph accuracy assessment, which models dependencies among triples by horn-clause coupling constraints [54], [55] and adopts a greedy algorithm with accuracy guarantee to iteratively choose a small set for human labeling. Later Gao *et al.* [56] provide an iterative sampling and evaluation framework for both static and evolving KGs with the thought of clustering, where various sampling strategies are proposed and compared, and the whole process is under a statistical framework with strong quality guarantee and minimal human efforts. They demonstrate by both theory and practice that to achieve a certain level of precision, the evaluation cost is only affected by the underlying KG quality rather than the size, showing the huge potential of sampling on large-scale knowledge graph quality assessment.

These works mainly focus on KG accuracy defined by the percentage of correct triples in the KG, but there are still large research gaps on other quality dimensions like completeness and redundancy being effectively evaluated by humans. And considering that humans make mistakes sometimes, crowdsouring technologies such as worker quality estimation and truth inference [57], [58] need to be taken into account.

3.2 Methods for Problem Discovery

Unlike quality assessment that can be implemented and estimated on a small sample set, the recognition of false assertions and error patterns requires in-depth perception and analysis on the concrete data. In [59], a generialized methodology for linked data quality analysis is proposed, comprising of a manual and a semi-automatic process. They focus on four quality dimensions and first define a category of 17 kinds of quality problems on DBpedia [1], and then a crowdsourcing tool, TripleCheckMate [60], is developed, where workers are allowed to choose and evaluate on individual resources, detect errors and link to the predefined error classification. [61] and [62] look deep into the effectiveness of crowdsourcing approaches on KG problem discovery. They combine two kinds of tasks: (1) a contest targeting at experts to find out and classify erroneous triples, and (2) microtasks published on the crowdsourcing platform Amazon Mechanical Turk[3], to be verified by ordinary workers. Their experiments show that these two paradigms of crowdsourcing can complement each other and crowdsourcing-based methods are promising to identify and address KG quality problems.

3.3 Methods for Quality Improvement

For those detected erroneous triples, one fixing way is to directly remove them from the database, which may lead to

3. https://www.mturk.com/

unnecessary loss of information. And missing facts need to be added for better completeness. Automated modification can give rise to new errors and thereby is often avoided in actual business scenarios [18], but quality improvement by humans is feasible and often necessary. In [63], Jiang *et al.* propose to use database facts and crowdsourcing verification for knowledge base enhancement on wrong and missing relationships between entities. They design a dynamic algorithm to select candidates within a limited budget and maximize the benefit, as well as techniques for graph updation according to the crowdsourcing results, where dependencies among triples are considered.

To conclude this section, quality management based on human is a natural and common way in industrial scenes but does not attract enough academic researches. People can easily participate in all aspects of quality controlment, but cost and scalability issues are quite severe for large-scale KGs. Sampling and crowdsourcing are two commonly used methods. How to guarantee accuracy with minimal budget and how to design suitable user interactions for different tasks still demand for more studies.

4 Methods Based on Statistics/Learning

This section discusses about methods based on statistics, from traditional statistical means like distribution-based outlier detection and classic machine learning algorithms, to the currently prevailing embedding-based representation learning and neural network methods.

4.1 Traditional Statistical Methods

4.1.1 Methods for Problem Discovery

Outlier means the one that appears to be inconsistent with the remainder of the dataset and outlier detection techniques have long been used to detect and remove anomalous data [64]. In KG quality management, these methods are helpful to identify errors, especially those numerical literal values, though sometimes they cannot well distinguish natural outliers from actual errors. In [65], Paulheim and Bizer exploit statistical distributions of properties and types to identify wrong triples and add missing type statements. They propose SDValidate, which assigns a confidence score to each statement and spots outliers by a given threshold. Experiments show that this method outperforms most previous works without extra knowledges. [66] uses different outlier detection methods like interquantile range and kernel density estimation, combining with various preprocessing strategies to identify numerical errors in DBpedia, reaches 87% precision and finds out 11 systematic errors to improve the construction process. [67] further exploits the *owl:sameAs* links between resources to alleviate the influence of natural outliers.

Following the outlier detection technologies based on statistical distribution, the thoughts of feature extraction and machine learning classfication are introduced to detect errors. [68] represents each link as a feature vector in a higher dimensional vector space and shows the effectiveness of outlier detection methods to identify wrong links between datasets. [69] extracts features by path kernels [70] and trains a binary classifier such as decision tree to conduct ontology reasoning and A-box consistency checking, proving the possibility of machine learning for approximate inconsistency detection. In [71], the outlier detection problem in numerical data is decomposed into a set of supervised learning problems, where a predictive model for each attribute is learned from other attributes to identify patterns as well as to derive weights and outlier scores. It is demonstrated that this method is robust to irrelevant attributes and is capable of giving concise explanations for outliers with symbolic methods. In [72], local path and type features are used to train a classifier for every relation to detect wrong relation assertions, which is further expanded in [73] to deduce higher level error patterns.

External resources are exploited in some works for fact validation and error detection. [74], [75], [76] search evidence from the web, text corpus and query logs, which are then used to judge the correctness of KG triples, and in this process technologies of knowledge fusion and truth discovery [77], [78] are adopted. Apart from these, [79] proposes several predicate matching functions to find identical resources from other knowledge graphs for validating RDF triples. And in [80], reference sets of similar entities are compared to identify unexcepted facts about entities.

Some graph exploration techniques are also proposed to discover errors in KGs. [81] devotes to detect wrong IsA relation in large-scale lexical taxonomies, which is modeled as the detection and elimination of cycles. They use two models based on DAG (directed acyclic graph) decomposition and level assignment respectively and give efficient algorithms. Another work is [82], aiming at discovering exceptional facts about entities in knowledge graphs. It models an exceptional fact as a context-subspace pair, and applies beam search as well as two heuristic algorithms to cope with the exponential search space, where the detected exceptional facts can be candidates of wrong and inconsistent triples. Both of these two works pay main attention on the scalability to handle very large graphs.

Besides, a knowledge graph triple trustworthiness measurement model is proposed in [83], which fuses characteristics of three levels from entity, relationship to KG global. For entity level, they propose an algorithm called ResourceRank to determine whether there is a possible relationship between entity pairs; for relation level, a translation-based energy function is used; and for KG level, a reachable paths inference algorithm is designed to measure the trustworthy of a given triple. These three features are combined together into a multi-layer perceptron to output the final triple values, which can detect incorrect triples in the graph.

Different from the above methods focusing on false assertions, [84] devotes to reveal common properties of the errors to explain where and how these errors happen in data generative process. They propose an error diagnosis framework, Data X-Ray, using feature hierarchies and Bayesian analysis to derive the most likely causes associated with the errors, and a top-down iterative algorithm as well as a parallel MapReduce version are implemented to scale to large datasets. Similarly, [73] derives higher level patterns from the errors by translating the classifiers of decision trees into SHACL-SPARQL relation constraints.

4.1.2 Methods for Quality Improvement

Similar to the methods for problem discovery, statistical distributions and external resources are used for knowledge graph quality improvement, expecially for completing missing information. In [85], a web-search-based question-answering technology is used to find missing objects for a given subject-relation pair on demand. They train the system by query logs and existing KB facts to learn what questions to ask in the web for different subject-relation pairs. [86] proposes SDType, using the statistical distribution of types in the subject and object of the property to predict the instance's missing type, which they believe has better tolerance with respect to noise in the data. [87] pays attention to errors derived from instance confusion (wrong links to entities with similar names) and attempts to correct them. They first perform error detection algorithms to find out potential false assertions, and retrieve candidate entities to do the replacement. The low accuracy is not satisfactory enough for practical correction on real-life KGs that the authors recommend to work as suggestions for users, which at the same time demonstrates the intractability of automatic error correction tasks.

In general, the trend of traditional statistics-based methods is moving from pure statistical distribution to explicit feature extraction and supervised machine learning, and more and more works try to absorb external resources and propose methods specific to graphs for KG problem discovery and quality improvement. The next direction is to substitute manual feature engineering with embedding techniques, which are discussed below.

4.2 Embedding-Based Methods

Motivated by the booming development of deep learning, the paradigm of graph mining and graph analysis is changing from traditional feature engineering to graph representation learning, by which the vertices, edges and subgraphs of a graph are converted into low-dimensional dense embeddings to be fed into various machine learning models for downstream applications. This is such a hot research direction with a large number of publications and surveys [31], [32], [88], [89] that we don't seek to cover all related works on graph embedding, but give an overview of how these methods can be used for knowledge graph quality management, with some classic embedding technologies presented and state the pros and cons of such embedding-based methods. To get a more comprehensive and in-depth understanding of embedding techniques, we recommend readers to read the surveys listed above.

4.2.1 Overview of Embedding-Based Methods

The key idea behind KG embedding is to learn the representation of graph components like nodes and edges into continuous vector spaces, with the structure and attribute characteristics being reserved. Early methods solely make use of the observed triples in knowledge graphs with various embedding spaces and learning models, and more and more works commit to introducing additional resources, such as entity types [90], textual descriptions [91] and logical rules [92], for better embeddings. And the learning model is getting more complex, from shallow distributed representation, to multilayer neural networks.

Typically, a KG embedding method can be decomposed into three steps [32]: (1) determining the representation space of entities and relations, where entities are usually represented as vectors, and relations are generally regarded as the operation between entities, represented as vectors, matrices, tensors, and so on; (2) defining a score function to capture features from the graph; and (3) designing a suitable model and corresponding algorithms to solve the optimization problem.

The output of the representation learning methods is a set of low-dimensional embeddings for different entities and relations, and one can access the accuracy of triples as well as inferring new facts by calculating on the learned embeddings. Typical evaluation and application tasks include link prediction [93], triple classification [91], entity classification [94] and open information extraction [95], all of which handle quality issues about completeness from different perspectives.

4.2.2 Graph Embedding Techniques: A Taxonomy

Here we talk about the main methods of knowledge graph embedding, which generally fall under four headings: translational distance models, tensor decomposition models, deep learning models, and models with additional information.

Translational Distance Models. This kind of models regard relations as geometric transformations in the vector space and measure the score of a fact by calculating the distance between the resulting vector after transformation and the tail entity. The most representative one of this kind is TransE [96], which is inspired by Word2Vec [120] and enforces that the embedding of the tail entity should be close to the sum of the head and relation embeddings for the right triples. It is simple enough to be trained on very large graphs and has been shown to be of effectiveness in many scenes, but cannot perform well on 1-to-N, N-to-1 and N-to-N relations. In order to overcome this disadvantage, many variants of TransE are proposed. For example, TransH [97] models a relation as a hyperplane together with a translation operation, thus enables an entity to have different roles in different relations. TransR [98] represents each relation as a dynamic mapping matrix, which is further simplified in TransD [99] and TranSparse [100]. Other works include TransM [101], CrossE [102], RotatE [103], and etc.

Tensor Decomposition Models. Models based on tensor decomposition represent the connections between nodes in the form of matrices or higher-order tensors, and obtain the node embedding by factorizing these tensors [88]. They vary in the representation space and the decomposition algorithm. RESCAL [104] represents each entity as a vector and each relation as a matrix, and the scoring function is computed as a bilinear product. It is simplified in DistMult [105], where all relation embeddings are restricted to be diagonal matrices and therefore the space of parameters is reduced. SimplE [106] learns two vectors for each entity and each relation like Canonical Polyadix (CP) decomposition [107], and enhances CP by capturing the dependency of the two vectors, which is fully expressive and is able to model asymmetric relations. HolE [108] employs circular correlation to create compositional representations, which can be seen as a compression of the tensor product and reduce the time and space complexity.

Deep Learning Models. Deep learning-based models adopt neural networks to capture the non-linearity in graphs, where parameters are organized into separate layers with different non-linear activation functions. Modern graph neural networks (GNNs) can address the embedding problem through a graph autoencoder framework [121], which typically use the connectivity and features of the graph and iteratively aggregate the node embeddings of local neighborhood.

There are many different kinds of GNNs depending on the various extraction and aggregation functions, including graph convolutional networks like GCN [109] and GraphSAGE [110], graph attention networks like GAT [111] and HGT [112], graph autoencoders [113], [114] and graph spatial-temporal networks [115], [116]. And inspired by the development of pre-trained models in natural language processing (NLP) and computer vision (CV), many self-supervised learning methods and pre-trained GNNs are proposed for better transferring among datasets and tasks [117], [118].

Models With Additional Information. In addition to the evolution of models and algorithms, many researches attempt to integrate more external resources, from textual information to logical rules. [119] presents a comprehensive survey about KG embedding techniques with additional literals such as text and numerical values. DKRL [94] learns the semantics of entity descriptions by two encoders, continuous bag-of-words and deep concolutional neural models, which are associated with the TransE [96] embedding for each entity to handle the zero-shot scenarios. [92] proposes a general framework to jointly model triples and rules, where rules help to get better entity and relation embeddings for tasks like link prediction, showing the effectiveness of joint learning.

Though embedding-based methods have shown great potential in efficiently mining and analysing on large-scale graphs and have many successful applications even beyond in-KG quality management, however, they still face many deficiencies. In terms of knowledge graph quality management, the problems can be solved by embedding are limited. Embedding-based methods can predict and complete missing information, such as entity types and relations, and sometimes help to identify redundant entities with other resources [122], but they cannot assess the overall KG quality. They learn from data and make predictions accordingly, assuming most (or all) of the input data is correct, which departs from our setting where the input KB has many quality issues, leading to the fact that they don't have good abilities to find and correct wrong triples. Experiments have shown that KG embeddings are quite sensitive to sparse and unreliable data [123]. And [124] finds that reverse triples and other redundancy in existing benchmarks lead to a substantial over-estimation of the embedding models' accuracy, and argues that link prediction doesn't have truly effective automated solution. These works remind us to reconsider the effectiveness of embedding methods on real-life KGs. Besides, embedding-based methods typically encode entities and relations into vectors, but overlook the semantics and dependencies of literal values. Apart from these, the poor interpretability and complex parameter selection of these methods are always mentioned. And the problem of transferring across different datasets and tasks has not been fully resolved yet. In summary, embedding-based methods are potential solutions for knowledge graph quality management, but they are far from enough and still have a long way to go.

5 Methods Based on Rules

Unlike the above two categories of methods that need different designs for different tasks, rules can put all quality management processes into a unify framework, which typically contains four steps: the definition of rules, rule extraction, rule assessment and evaluation, rule application to conduct problem discovery and quality improvement. Various forms of rules have been proposed together with mining and application algorithms, all of which have to make a balance between expressivity and computational complexity. Next we first present an overview of rule-based methods and move on to several types of rules for KG quality management.

5.1 Overview of Rule-Based Methods

As a classic symbolic reasoning technique, rule and rule learning have a long research history, from early inductive logic programming [167], to those studied in relational data mining in databases [168], and rules in KGs [29]. Rules represent knowledge in an explicit way and can be enhanced with human intelligence easily, which makes it keep an important position even in the era of deep learning and neural networks. Manual writing is the most direct way to generate rules, but it is diffcult for humans to exhaust all. Rule learning makes it possible to automatically discover rules and has become an important subfield of machine learning.

Rule learning algorithms can be roughly divided into two groups, those frequent pattern mining methods that aim to discover typical patterns from the dataset to be transformed into corrseponding rules, e.g., [126], [127], and those enhanced by embedding techniques for efficiency and accuracy [133], [134]. It often comes with a rule assessment stage with some statistical metrics like support and confidence in an automatic rule mining process (see [169] for more metrics). And in case of KGs where issues like inaccuracy and incompleteness widely exist, new evaluation metrics are being proposed, such as those in [170], [171]. Many works suggest to check the extracted rules by domain experts before applying, and a latest work [172] introduces the thought of human-in-the-loop and designs a few-shot knowledge validation framework for interactive quality assessment of rules, which takes the rule validation forward one step.

Rules can be directly used to discover and correct quality issues like errors and incompleteness in KGs, but often face the problem of efficiency and scalability, which is explored in different researches.

5.2 Predicate Logic Rules

First-order predicate logic rules are main reasoning methods in early statistical relational learning field [28] and have been widely studied in inductive logic programming (ILP), e.g., [173], [174]. Classical ILP systems usually cannot be applied to KGs due to the open world assumption and the scalability problem. And more and more successful methods have been proposed in KGs. Horn rule, a proper subset of First-order predicate logic, is the most commonly adopted rule form. It is a formula of the form $B_1 \wedge ... \wedge B_n \Rightarrow H$, where $B_1 \wedge ... \wedge B_n$ is a set of body atoms showing conditions and H is the head atom. Here each atom is a relation pair *r(X,Y)*, meaning that there is a relation r between entities X and Y. For example, bellow is a horn rule, which means that if Y is the daughter of X and Z is the wife of X, then Y is also the daughter of Z.

$$daughter(X,Y) \wedge wife(X,Z) \Rightarrow daughter(Z,Y)$$

AMIE [48] is a classic rule mining system for horn rules on large RDF knowledge bases based on the *partial completeness assumption* (PCA). It defines a set of mining operators and explores the search space by iteratively extending rules. And a suite of optimization strategies are proposed in AMIE+ [125] and AMIE 3 [126] to further speed up the rule mining process. The mined rules are used to predict missing relations in KGs, i.e., for completeness improvement, with satisfactory accuracy. But they remove all facts with literals (such as attribute and type information) and thus the expressivity and applicability of the rules are limited.

RDF2Rules [127] is another rule learning method, which generates rules by searching for frequent predicate cycles evaluated with a confidence score. Compared to AMIE+, it takes additional advantage of entity type information for accuracy and runs faster. But it still faces the problem of expressivity and applicability as AMIE does. [128] devotes to revise the given horn rules by adding negated atoms into the bodies, which enhances the ability to catch possible exceptions to some extent.

RuDiK [129], [130] models the rule mining process as an incremental graph exploration problem and adopts the A^* graph traversal algorithm [175] to get the most promising path at each iteration. It reconsiders the open world assumption and presents a generation algorithm of negative examples to mine rules over erroneous and incomplete KBs. Both positive and negative rules are mined from RuDiK, where the former can identify missing relationships between entities, and the later helps to detect errors and contradictions. Besides, it allows literal comparisons and constant selections in the rules, which enables much more patterns to be expressed.

[131] is a pioneering work to learn completeness assertions for relations in KGs, which can be used to measure the fine-grained completeness for single predicates, as well as to identify missing relations and improve the precision of fact prediction. They propose a set of signals indicating completeness of properties and by combining and injecting these signals into AMIE, they obtain high-quality completeness rules which can achieve a up to 100% precision for some relations on real KBs.

Meilicke *et al.* strongly emphasize the advantages of rule methods in KG completion tasks and propose AnyBURL [132], an efficient bottom-up rule learning system for uncertain horn rules by exploring KG paths. The rules are used to predicate missing objects with reusability and interpretability, and the algorithm is proved to be faster than previous systems.

Apart from these, more and more works try to learn logic rules with the help of embedding techniques. RLvLR [133] reduces the task of rule learning into that of searching for plausible paths of predicates. It proposes a new sampling method to start from the target predicates and path rules are extended iteratively by using the embeddings of the sample graph, which are then evaluated and pruned according to some novel scoring functions. It is demonstrated that RLvLR is faster and is able to mine more quality rules than AIME+, and outperforms Neural LP [176] in terms of efficiency and accuracy in link prediction tasks. In [134], rule learning is guided by a precomputed embedding model and external information sources like text corpora, where the efficiency and link prediction precision are improved. DRUM [135], an end-to-end differentiable rule mining system, adopts bidirectional RNNs to learn rule structures and scores simultaneously, where the learned rules are used for knowledge graph completion.

Although embedding techniques have shown potential to assist logic rule learning for efficiency and accuracy, most of these works are limited to predicate paths, where the rules are able to do link prediction and complete missing relations in an interpretable way, but not capable of identifying errors and handling literal facts. How to exploit the embedding methods to learn more complex rules for more tasks still needs a lot of exploration and endeavor.

What's more, RuleHub [136] aims to build an extensible corpus of rules for public KGs, where the rules are learned by existing methods like AMIE and RuDiK, and evaluated by both statistical metrics and human beings. These rules can be used as metadata and help to manage quality for public KGs. And [137] proposes a human-in-the-loop rule learning approach, where a GAN-based method is used to learn a confidence score for each rule, and a game-based crowdsourcing framework is devised to refine the rules, showing means to combine machine and human intelligence in rule learning.

5.3 Ontology Rules

In the context of Semantic Web, ontology reasoning and RDF validation have been studied for a long time and many constraint languages have been designed, which can be used as constarint rules for quality management. [177] gives a clear classification that such languages can be either existing frameworks like the RDF query language SPARQL [178] and the web ontology language (OWL) [179], or specific languages only designed for validation, such as SHACL [180] and ShEx [181]. And their execution is based on either reasoning or querying frameworks.

[182] proposes 81 types of constraints for various data applications, studies the role of reasoning for each constraint type, and compares the expressivity of five commonly used constraint languages. And [183] is another survey to overview and compare the characteristics and expressiveness of different RDF validation languages, hoping to point out directions for further development of such constraint languages. Based on these studies, Meester *et al.* [138] present a rule-based reasoning framework for RDF validation, which relies on N3Logic and EYE reasoner, and is independent of constraint languages. It can identify

constraint violations and generate root cause explanation, which helps to discover false assertions as well as error patterns in KGs. In [139], [140], DBpedia ontology is aligned to the foundational ontology DOLCE-Zero, which is then used to reason on the graph and cluster conflicts for identifying systematic errors. And [141] adopts description logic axioms as constraints and learns to fix constraint violations from the edit history of KBs, and their evaluation on Wikidata shows significant improvement.

As for the querying framework, SPARQL language is usually used as the constraint rules. In [142], an ontology-based data quality management architecture is proposed, where generic SPARQL query templates are defined to discover data quality problems including syntax errors, missing types, unique value violations, value range excess and functional dependency violations. Further more, Kontokostas *et al.* [143] create a set of 17 Data Quality Test Patterns (DQTP) to cover common quality issues according to their analysis on DBpedia, which can be instantiated into SPARQL queries and tested on the RDF dataset. They adopt these queries to evaluate five LOD datasets and reveal many quality issues. This work is then integrated into a platform called RDFUnit[4], where test cases can be created in five ways: changing from RDFS/OWL constraints, enriching constraints by tools like DL-Learner [184], re-using tests based on common vocabularies, instantiating existing DQTPs and writing own DQTPs. This set of templates points out common types of KG errors and can be directly used by running SPARQL queries, and at the same time serves as an important bridge between machine learning and domain experts, which is a potential method for KG quality assessment and problem detection.

5.4 Graph-Pattern Rules

In recent years, more and more works start to propose rules dedicated to graphs, where graph patterns are often included in rule bodies.

Following the research paradigm of dependencies in relational data, Fan *et al.* propose graph functional dependency (GFD) [144] and a set of extensions [148], [149], [150], [151], [152], which provide means to specify the semantics of the schemaless graphs and help to identify and correct quality issues. These graph dependencies are defined in the form of $Q[x](X \Rightarrow Y)$, where $Q[x]$ is a graph pattern, and X and Y are conjunctions of literals of x. A basic GFD has two types of literals: constant literal x.A = c, where c is a constant and A is an attribute except for id, and variable literal x.A = y.B, where A and B are attributes that are not id. Literals are extended in GED [147], [148] to support id literals to express keys, and in NGD [149] to contain linear arithmetic expressions and built-in comparison predicates. And in [152], timestamps are associated to the variables to specify the time span, which forms into TGFD to handle temporal graphs. This kind of definitions combine classic attribute dependencies with topological structures of graphs, able to express dependencies like those in relational data, and at the same time deal with the heterogeneity and flexibility of graphs.

Graph dependencies have stronger expressive power in comparison with those in relational data, and it is harder to mine and reason as well, which is studied in detail in [145], [148]. A parallel scalable algorithm for discovering GFDs is developed in [146], which combines pattern mining and functional dependency discovery in a single process and provides effective pruning strategies, showing the feasibility and scalability to find frequent and reduced GFDs in large graphs. The extracted graph dependencies are capable of capturing various semantic constraints on graphs, and can be used to detect errors and inconsistencies as well as completing and correcting issues in terms of KG quality management. [144], [151] and [149] explore parallel and incremental algorithms to detect errors and [150] attempts to deduce certain fixes based on the rules with the assistance of user interaction. This line of research has solid theoretical foundation, but actually effective algorithms are just getting started.

Apart from these graph dependencies, association rules with graph patterns (GPARs) are proposed in [153], which aims to mine frequent patterns in the graphs that can be used to predict missing relations. Its semantics are not sufficient to handle other quality issues, which are extended in [154]. In that paper, association rules, graph functional dependencies and even machine learning classifiers are incorporated into a uniform framework, which makes use of both rule-based and ML-based methods and is able to capture and solve incomplete and inconsistent information. And [155] develops graph temporal association rules (GTARs) to capture temporal associations of complex events.

Some rules focusing solely on graph-repairing are studied. Neighborhood constraints (NC) are used to detect and repair vertex labels in [156], [157], where several approximate algorithms are proposed to solve the NP-hard problem. [158] and [159] consider three kinds of repairing semantics including incompleteness, conflicts and redundancies, and design the Graph-Repairing Rules (GRRs) with corresponding repairing algorithms, which have similar structures with GFD while adding more literal types to carry out more kinds of repairing operations.

What's more, Belth *et al.* [160] adopt the idea of compression in information theory that compression techniques can find patterns in data and in turn reveal anomalies. They therefore use labeled and rooted summarization graphs as soft rules and build a system called KGist, to show what is normal and then identify strange and missing information. They learn the summarization based on the Minimum Description Length principle, and experiments of error and incompleteness identification on real KGs demonstrate the efficiency and effectiveness of such rules.

5.5 Other Rules

In addition to the above methods, there are some other rules for KG quality management.

In [161], class hierarchy is used to automatically determine obligatory attributes in graphs, which can be used to assess the completeness, identify missing information and help to improve the coverage. [162] presents RDFind, a distributed system to discover conditional inclusion dependency (CIND)

4. http://rdfunit.aksw.org

in RDF data. Though not directly applied to quality management, CIND is capable of describing inclusion semantics and identifying quality issues. In order to speed up the process of detecting and explaining inconsistency in large KGs, [163] gives an abstraction-based framework to find ontology rules on the partitioned graph modules, where their main focus is on splitting and summarying the graph and identifying inconsistency explanations from local modules.

Some works pay attention to uncertain knowledge bases, where Markov Logic Networks (MLNs) and probabilistic soft logic (PSL) are often used. [164] presents ProbKB, a probabilistic knowledge base which uses a relational DBMS to infer missing facts by an efficient SQL-based inference algorithm. [165] uses a numerical extension of MLN and a set of Datalog constraints to detect inconsistencies in uncertain temporal knowledge graphs (UTKGs), and a maximum a-posteriori inference (MAP) is carried out to get a most probable and conflict-free temporal KG. And TeCoRe [166], a system for temporal inference and conflict resolution in UTKGs is developed, where domain experts specify rules and constraints to be reasoned by several MLN and PSL solvers to detect and remove noisy temporal facts. These works need more effort on the problem of scalability and the automatic derivation of rules.

In summary, rule is a classic and enduring symbolic resoning technique and various types of rules have been proposed and adopted in KG quality management. Rules are highly interpretable and partially reusable, able to absorb human intelligence and transfer among different datasets. They can identify and solve various quality issues in a uniform framework with high precision, and such methods generally don't require to learn dataset-specific hyper parameters, showing advantages different from statistical methods. However, it is always a challenging task to obtain useful rules. No matter by manual writing or machine mining, the collected rules are hard to be complete and probably to cover only a subset of patterns in KGs. And there is always a trade-off between expressivity and complexity of the rules. How to select appropriate rule form with tolerable time and space overhead for different task requirements remains to be further explored.

6 Hybrid Methods

Here we talk about hybrid methods, where more than one technique of human intelligence, statistical/learning means and rule reasoning play an important role.

6.1 Human & Statistics/Learning

Due to the intractability of many realistic problems, human intelligence is often introduced in automatic methods and the technique of human-in-the-loop is helpful in the entire machine-learning pipeline [185]. In terms of KG quality management, methods combining human and statistics are showing great potential.

In [186], [187], a human and machine cooperation framework, HUMO, is proposed for entity resolution, which aims to divide the workload between human and machine such that a given quality requirement can be met with minimal cost. They present three optimization approaches based on monotonicity assumption of precision, sampling and hybrid techniques, which solve the problem of entity redundancy and are potential to be applied to other quality issues. [188] proposes a human-in-the-loop outlier detection approach, where humans are introduced to check the candidate outliers generated by the unsupervised algorithms. To discover all outliers with minimum human efforts, clustering and question selection methods are adopted. Based on whether noisy type labels and additional annotations are used, [189] categorizes typing error detection methods into four paradigms and experimentally shows that semi-supervised noise models are the most feasible and effective solution. They combine a neural network architecture with a probabilistic noise model for the type error detection task, where an active learning algorithm is used to iteratively get human-verified gold labels and the learning-rate is dynamically adjusted.

6.2 Human & Rules

There is a natural connection between rules and humans that rules can be created and examined by human experts, but this interaction is long neglected in researches. [137] presents a human-in-the-loop rule learning framework with high coverage and high quality, where candidate rules are first generated by machine algorithms and evaluated by a GAN-based method to get a confidence score, and then a game-based crowdsourcing framework is devised to refine the rules. It also tries to sovle the possible conflicts when using various rules. In [172], embedding techniques and user feedback are used interactively to assess the quality of a particular rule, which leads to better estimation of the confidence score than simple statistical measures.

Human and rules can also work together in a quality management process. Arioua and Bonifati present a user-guided KB repairing method based on update in [190], where tuple-generating dependencies (TGDs) and contradiction detecting dependencies (CDDs) are used as the logic rules to give candidate repairing suggestions, and the final repairing strategy is further guided by user interaction, by which means the repairing can be implemented semi-automatically to meet the user's requirements.

6.3 Statistics/Learning & Rules

As two classic reasoning methods, there is a two-way interaction between statistics and rule techniques. On the one hand, rule learning often comes with statistical metrics to define the confidence, and many embedding methods are adopted to guide the rule learning process [133], [134]. On the other hand, rules can serve as additional resources to assist embedding learning [92]. And apart from these interactions, statistics and rules can have more diverse ways to promote and complement each other in processes of knowledge graph quality management.

In [191], a fine-grained evaluation for knowledge graph completion is conducted on several rule- and embedding-based systems, where the test sets are partitioned by involved rule types. Experiments show that both rule- and embedding-based methods have problems in solving certain types of completion tasks and an ensemble learning method is proposed to combine these two families of approaches, where ensemble weights are learned for each realtion to fully utilize the advantages of different methods on different tasks. [192] attempts to correct erroneous assertions of entities and literals with the help of both deep learning and rule reasoning. In this framework, multi-relational sub-graphs are extracted according to semantic relatedness for each identified wrong assertion, where a link prediction model is learned by both semantic embeddings and observed path and node features to predict possible substitutions. And semantic consistency checking with property range and cardinality constraints is then conducted to help to make the final correction decision. Experiments on two datasets have shown the effectiveness of this framework on both general and enterprise KGs.

[193] proposes a new iterative framework to learn embeddings and rules at the same time and make their advantages complement to each other. This framework contains three parts: (1) embedding learning based on existed triples and those inferred by rules; (2) rule learning assisted by embeddings; and (3) axiom injection to add new triples derived by rules into KGs. These steps are conducted iteratively during training. In this process, rules and the injected triples help to improve the embedding quality on sparse entities, embeddings assist to learn more quality rules more efficiently, and the whole performance to complete missing facts is improved, which shows the huge charm of interactive learning.

In [154], rule-based and machine learning-based methods are further unified, where embedding-based ML classifiers are incorporated into the rules as predicates, i.e., a ML classifier here becomes part of the rule itself. They define graph association rules (GARs) that have similar semantics like GFDs [148], and any well-trained ML classifiers can be added into the literal constraints. Theoretical analysis and parallel incremental inference algorithms are well studied, showing a new direction to combine rule and ML techniques.

6.4 Human & Statistics/Learning & Rules

Hao *et al.* [194] together use machine learning, human-in-the-loop approach and logic rules to detect outdated facts in KGs. In this framework, a binary classifier is trained with features like historical update frequency and time span to predict the likelihood of each fact being outdated, then verification is conducted by humans and the human answers are further expanded by logic rules to get new facts, which are added back to the machine classifier. The processes move on iteratively until the accuracy requirement is met.

To sum up, each family of human-, statistics- and rule-based methods has deficiency and limitation for knowledge graph quality management, and their combinations are showing great potential with many researches proposed. How to design more delicate methods to make full use of these different technologies remains an open question worth exploring.

7 Discussions and Future Work

To give readers an overall perspective about the literature, we plan to conclude some key issues and propose several future directions here. Specifically, in Section 7.1 we discuss about what is focusing on and what is missing in existing works, and in Section 7.2 we recommend what is next.

7.1 Discussions

7.1.1 Technologies

Human can take part in all processes of knowledge graph quality management with high precision and interpretability, and human-based methods are quite common in practice. But due to the scale of real-life KGs, it is impossible for humans to check all facts and purely artificial methods do not arouse much academic interest. Recent works start to study the sample framework to assess KG quality with accuracy guarantee and acceptable cost, showing a great potential in both theory and practice. And crowdsourcing technologies are introduced to identify and correct KG issues. But these works mainly focus on dimensions of accuracy and sometimes completeness that other dimensions with delicate user interactions remain to be explored.

Statistics-based methods are listed in Table 1, which displays main works with not only processing goals, target dimensions and typical techniques, but also the object types like relation and attribute, and the column of resources is used to show whether external information is introduced.

We can conclude that, from traditional outlier detection and classic machine learning algorithms, to those various embedding-based techniques that gradually replace manual feature engineering, the statistics-based methods have shown their efficiency and strong ability for KG quality management, especially in error detection (e.g., outlier detection techniques) and graph completion (e.g., link prediction by embedding methods). And the embedding-based deep learning and neural networks are playing important roles in more fields than quality management.

However, it can be seen from the table that most of these works are limited in tasks of error detection and graph completion with dimensions of accuracy and completeness, showing their deficiency in handling various problems and dimensions. Statistics-based methods learn from the data, which may perform poor when the dataset has many quality issues such as errors and sparsity, and therefore how to incorporate external resources into such methods is becoming a popular direction. And most of these works (especially those embedding-based ones) focus mainly on relations and overlook the semantic dependencies of attributes and literal values, which is worth further exploring. What's more, this type of methods is known for poor interpretability and complex parameter selection, and thus further research is required on model interpretation and transferring learning.

Rule-based methods are concluded in Table 2. This table is organized according to the four-step framework of rule definition, rule extraction, rule assessment and rule application with corresponding works and techniques. And for rule application, target goals, dimensions and objects are also listed. The elements with placeholder '/' mean the issues not paid attention to by the papers.

From Table 2, it is clear that rule-based methods have attracted much research interest with various forms of rules as well as extraction, evaluation and application techniques being proposed. Most researches on predicate logic rules pay main attention to efficient rule mining algorithms and

TABLE 1
Summary of Statistics-Based Methods

Paper	Goals	Dimensions	Objects	Techniques	Resources
[86] (2013)	GC	CP	ET	statistical distribution	KG
[65] (2014)	ED, GC	A, CP	ET, R	statistical distribution	KG
[66] (2014)	ED	A	NA	outlier detection methods	KG
[67] (2014)	ED	A	NA	outlier detection with external links	KG, sameAs links
[68] (2014)	ED	A	L	outlier detection with feature engineering	KG, sameAs links
[85] (2014)	GC	CP	R	web fact validation	KG, web, query logs
[71] (2015)	ED	A	NA	supervised machine learning	KG
[74] (2015)	ED	A, T	R, A	web fact validation	KG, web
[80] (2015)	ED	A	A	outlier detection with external resources	KG, reference entity sets
[84] (2015)	EPD	A	ET, R, A	Bayesian analysis	KG
[69] (2016)	ED	A, CS	ET, R, A	supervised machine learning	KG, ontology
[72] (2017)	ED	A	R	supervised machine learning	KG
[76] (2017)	ED	A	R, A	web fact validation	KG, web, query logs
[79] (2017)	ED	A	A	semantic matching	KG, external KGs
[81] (2017)	ED	A	R(IsA)	graph based models	KG
[87] (2017)	EC	A, R	E	supervised machine learning	KG, external links
[82] (2018)	ED	A, CS	R, A	graph exploration	KG
[75] (2019)	ED	A	R, A	web fact validation	KG, web, text corpus
[83] (2019)	ED	A	R	neural network	KG
[73] (2020)	ED, EPD, EC	A	E, R	supervised machine learning	KG, external links
[96], [97], [98], [99], [100], [101], [102], [103], etc.	GC	CP	ET, R	embedding (translational distance model)	KG
[104], [105], [106], [107], [108], etc.	GC	CP	ET, R	embedding (tensor decomposition model)	KG
[109], [110], [111], [112], [113], [114], [115], [116], [117], [118], etc.	GC	CP	ET, R	embedding (deep learning model)	KG
[92], [94], [119], etc.	GC	CP	ET, R	embedding (with additional information)	KG, external resources

Abbreviations used: Goals (A = Assessment, ED = error detection, EPD = error pattern derivation, EC = error correction, GC = graph completion), Dimensions (A = accuracy, CS = consistency, CP = completeness, T = timeliness, R = redundancy), Objects (E = entity, ET = entity type, R = relation, A = attribute, NA = numerical attribute, L = link).

the most used evaluation strategy is statistical measures, except that [136] and [137] introduce humans to assist the assessment. Closed horn rule is the most adopted rule form for predicting missing relations. Some attempt to enhance the expressivity for more tasks, like [129] to discover both positive and negative rules with literal comparisons so that error detection and attribute issues can also be handled. The mined rules are used directly, where the complexity and possible conflicts during the application is omitted by these works and further research needs to be done.

Ontology rules generally show more powerful expressivity to cover various quality dimensions. Both reasoning- and querying-based methods mainly commit to defining an integrated error detection framework with existing constaint languages or formulated query templates, and the rules are usually obtained by human-written, where [141] to mine from edit history and [143] to learn from data make some progress. Besides, it remains to be explored that how these ontology rules can be used for other tasks like graph completion and error correction.

Graph-pattern rules incorporate graph structures into the rule body, which are capable of expressing complex semantics and coping with more tasks. As technologies dedicated to graphs, they are attracting more and more attention in recent years, and various types of rules are proposed with different expressivity for different task requirements. The main extraction technique for these rules is to discover frequent graph patterns with statistical measures, and reasoning is used to check and reduce the mined rules. Though in works of graph dependencies it is suggested to check the rules by experts before actually using, the interaction is not studied yet. Many of these works notice the complexity of rule application, and they have made some attemption on parallel or incremental algorithms for large-scale knowledge graphs. Graph-pattern rules are showing huge potential in KG quality management, but they are still in the initial stage, where more effective extraction, evaluation and application algorithms need to be proposed, and a general rule form with flexible and optional complexity is to be determined.

Besides, other kinds of rules such as probabilistic soft logic are also studied for KG quality management. And they also face the problems of rule extraction, trade-off between complexity and expressivity as well as scalability on large graphs.

Hybrid methods are shown in Table 3. As the advantages and disadvantages of different types of methods summaried in Table 4, hybrid methods have the ability to combine the strengths of different techniques and many

TABLE 2
Summary of Rule-based Methods

Rule Type		Paper(s)	Rule Extraction	Rule Assessment	Rule Application			
					Goals	Dimensions	Objects	Techniques
Predicate Logic Rules		AMIE [48], [125], [126]	operator expansion	statistical measures	GC	CP	R	/
		RDF2Rules [127]	frequent pattern mining	statistical measures	GC	CP	R, ET	/
		[128]	revision of learned rules	statistical measures	GC	CP	R, ET	/
		RuDiK [129], [130]	incremental graph exploration	statistical measures	ED, GC	A, CS, CP	R, A	/
		[131]	operator expansion	statistical measures	A	CP	R	/
		AnyBURL [132]	graph path exploration	statistical measures	GC	CP	R	/
		RLvLR [133]	sample & embedding	statistical measures	GC	CP	R	/
		[134]	embedding & text corpus	statistical measures	GC	CP	R	/
		DRUM [135]	neural networks	end-to-end	GC	CP	R	/
		RuleHub [136]	existing methods like AMIE	statistics & human	ED, GC	A, CS, CP	R, A	/
		[137]	machine learning	machine & human	ED, EC, GC	A, CS, R, CP	E, R	label aggregation methods
Ontology Rules	reasoning-based	[138]	given constraints like SHACL	/	ED, EPD	A, CS	R, A	logic reasoning
		[139], [140]	ontology mapping	/	EPD	A, CS, CP, R	R, A, ET	clustering
		[141]	mining from edit history by AMIE	statistical measures	EC	A, CS, CP	R, ET	sorting by confidence
	querying-based	[142]	instantiating templates by hand	human	ED	A, CS, CP	R, ET, A	sparql querying
		RDFUnit [143]	automatic/semi-automatic/manual	human	ED	A, CS, CP, R	R, ET, A	sparql querying
Graph-pattern Rules	graph dependencies	GFD [144], [145], [146]	frequent pattern mining	reasoning & human	ED	A, CS, CP	R, ET, A	parallel subgraph matching
		GED [147], [148]	frequent pattern mining	reasoning & human	ED	A, CS, CP, R	R, ET, A, E	parallel subgraph matching
		NGD [149]	frequent pattern mining	reasoning & human	ED	A, CS, CP	R, ET, A	parallel incremental algorithm
		GFix [150]	frequent pattern mining	reasoning & human	ED, EC	A, CS, CP, R	R, ET, A, E	parallel chasing algorithm
		GDD [151]	frequent pattern mining	reasoning & human	ED, EC	R	E	algorithm for entity resolution
		TGFD [152]	frequent pattern mining	reasoning & human	ED	A, CS, CP, T	R, ET, A	parallel incremental algorithm
	graph association rules	GPAR [153]	frequent pattern mining	statistical measures	GC	CP	R	parallel algorithm
		GAR [154]	frequent pattern mining & ML	statistics & reasoning	ED, GC	A, CS, CP, R	R, ET, A, E	parallel incremental algorithm
		GTAR [155]	event & frequent pattern mining	statistical measures	GC	CP, T	R	subgraph matching
	graph repairing rules	NC [156], [157]	neighborhood distribution	statistical measures	ED, EC	A, CS	ET	approximate algorithms
		GRR [158], [159]	frequent pattern mining	reasoning	EC	A, CS, CP, R	T, E, A	decomposition-and-join strategy
	graph summarization	KGist [160]	graph summarization	MDL principle	ED, GC	A, CS, CP, R	R, ET, A, E	subgraph matching

TABLE 2
(*Continued*)

Rule Type	Paper(s)	Rule Extraction	Rule Assessment	Rule Application			
				Goals	Dimensions	Objects	Techniques
Other Rules	[161]	statistical model	statistical measures	A	CP	A	/
	RDFind [162]	frequent pattern mining	statistical measures	/	/	/	/
	[163]	input ontology	/	ED, EPD	CS	R, ET	abstraction & reasoning
	ProbKB [164]	input rules	/	GC	CP	R	parallel sql-based inference
	[165]	input Datalog constraints	/	ED, EC	CS, T	R	MLN reasoning
	TeCoRe [166]	expert input	human	ED, EC	CS, T	R	MLN reasoning

Abbreviations used: Goals (A = Assessment, ED = error detection, EPD = error pattern derivation, EC = error correction, GC = graph completion), Dimensions (A = accuracy, CS = consistency, CP = completeness, T = timeliness, R = redundancy), Objects (E = entity, ET = entity type, R = relation, A = attribute).

researches are proposed for various combination frameworks in recent years. It is an ascendant and potential field to be further explored.

7.1.2 Goals

Looking back into the previous tables, one interesting finding is that most works focus only on tasks of error detection and graph completion, while other quality management processes like assessment, error pattern deduction and error correction are generally overlooked. Quality assessment is a necessary step to quantify the fitness of KGs for downstream tasks and for further improvement. A few explorations have been done by manual sampling [53], [56] and rules [131], [161], but the problems of granularity and various dimensions remain unsolved. Error pattern deduction helps to find out the causes of errors so that the KG quality can be improved from the source. Statistics- and rule-based methods have a little preliminary attempt on this task [73], [84], [140] and more work is required. Error correction is long recognized as an intractable problem even in relational data. As it is possible to introduce new errors in the correction process, purely automatic methods are often avoided. Human can play an important role in this task and we are pleased to see more and more rules and hybrid methods are proposed to cope with it.

Additionally, most of the works pay attention to only one or two goals and few of them try to give an overall framework to cover all problems. We say that rule-based methods have such ability, but out of the complexity they typically focus on one thing. Further study on a flexible and unify overall solution is therefore recommended.

As for the target objects, relation has the most focus, while attributes, expecially the semantic dependencies in literal values are required to be further studied.

7.1.3 Dimensions

Though we have tried our best to search for works on different kinds of quality dimensions, it has to be admitted that accuracy and completeness have attracted the most attention. Besides, outlier detection techniques and expressive rules can identify inconsistent facts in the graphs so that the consistency issue is being solved gradually. But there are still many unanswered questions about redundancy and timeliness. A possible explanation for the lack of researches on timeliness issues is that KGs with rich time information are not common at present. And redundancy is often

TABLE 3
Summary of Hybrid Methods

Paper	Goals	Dimensions	Objects	Type	Techniques
[186], [187] (2018)	ED	R	E	H-S	workload distribution with quality assurance
[188] (2020)	ED	A, CS	NA	H-S	clustering and question selection methods
[189] (2021)	ED	A, CS	ET	H-S	neural network, probabilistic model, active learning
[137] (2018)	ED, EC, GC	A, CS, R, CP	E, R	H-R	machine learning, GAN, crowdsourcing
[172] (2021)	ED, GC	A, CS, CP	R	H-R	machine learning, embedding, interactive learning
[190] (2018)	EC	CS	R, A	H-R	human-machine interaction
[191] (2018)	GC	CP	R	S-R	ensemble learning
[192] (2020)	EC	A, CS	R, A	S-R	deep learning, consistency reasoning
[193] (2019)	GC	CP	R	S-R	joint learning framework
[154] (2020)	ED, GC	A, CS, CP, R	R, ET, A, E	S-R	machine learning, rule reasoning
[194] (2020)	ED	A, CS, T	R, A	H-S-R	machine learning, interactive learning

Abbreviations used: Goals (A = Assessment, ED = error detection, EPD = error pattern derivation, EC = error correction, GC = graph completion), Dimensions (A = accuracy, CS = consistency, CP = completeness, T = timeliness, R = redundancy), Objects (E = entity, ET = entity type, R = relation, A = attribute), Type (H-S = human + statistics, H-R = human + rules, S-R = statistics + rules, H-S-R = human + statistics + rules).

TABLE 4
Summary of Different Kinds of Methods

Methods	Advantages	Disadvantages	Representative Works
Human-based	high precision, high interpretability, flexible, easy to conduct in all processes	costly and time consuming, inexhaustible on large KGs	assessment [53], [56] error detection and correction [60], [63]
Stastics/ Learning-based	efficient, powerful reasoning ability	limited in task form and dimensions, sensitive to input data quality, poor interpretability and complex parameter selection, hard to transfer	distribution-based outlier detection [65], [67] classic machine learning [71], [73] embedding-based methods [96], [109]
Rule-based	high precision, high interpretability, reusable, simple parameter setting, easy to absorb human intelligence	hard to extract rules, difficult to cover all patterns, lack of uniform rule form	predicate logic rules [126], [129] ontology rules [138], [143] graph-pattern rules [144], [160] other rules [161], [165]

considered as the task of entity disambiguation in the process of KG construction and fusion, so the detection and elimination of redundancy on built KGs are not talked much. Although inaccuracy and incompleteness are indeed the two most common KG quality problems, more research should be undertaken to investigate other dimensions to present a comprehensive view of knowledge graph quality management.

7.2 Future Work

The findings above show a number of important implications for future practice, which are listed below.

Overall Solution. The ultimate goal of quality management is to get satisfactory knowledge graphs, which lies in three progressive processes of assessment, problem discovery and quality improvement, as well as various dimensions and objects. However, most current works focus only on some of the process goals and dimensions, and therefore a flexible and configurable framework for all kinds of task requirements is in urgent need.

Various Dimensions. Knowledge graph quality issues go beyond inaccuracy and incompleteness. And thus other dimensions like timeliness and redundancy need to be paid more attention to.

Objects Beyond Relations. Regardless of human-, statistics-, rule-based or hybrid methods, we can see that relation is the most targeted object. However, literal attributes also play an important role in KGs. How to identify and correct literals with rich semantics and infinite values by automatic methods remains a cool research area.

Human Participation. Humans are the initial source and ultimate beneficiary of knowledge, and manpower plays a significant role for quality management in industrial scenes. But academic researches on this are not enough. How to introduce human intelligence in more ingenious ways for more tasks is still lacking.

Combination Strategies. Different types of techniques have their own advantages and disadvantages, where hybrid methods can make them complement to each other. For example, rules and human intelligence may be potential supplements to improve the interpretability of deep learning models. It remains to be further studied for various combination strategies, especially a framework to put humans, statistics and rules together.

External Resources. As there are potential quality issues in the input KGs, learning purely from the data is prone to be misled. Therefore how to incorporate external information and knowledge to correct the deviation is a rising and promising direction.

Efficiency and Scalability. Although various researches have been done to cope with large-scale knowledge graphs, the problem of efficiency and scalability is by no means solved. More studies on actually usable algorithms, such as parallel, incremental, or approximate strategies, are still required.

Dynamical Knowledge Graphs. Most existing works focus mainly on static graphs. However, real-life KGs often evolve with time. As the emerging of more and more temporal knowledge graphs [195], quality management on dynamical KGs may become a research hotspot in the future.

Taken together, knowledge graph quality management is a comprehensive research topic covering various tasks, dimensions and objects. And there is an internal integration of data, rules, manpower and learning models behind the means. No matter focusing on dedicated or general methods, static or dynamic graphs, theory or application algorithms, there is broad research space worth exploring.

8 Conclusion

In this paper, we present a comprehensive survey on knowledge graph quality management, from basic concepts of quality issues, dimensions and metrics, to various works on different quality management processes. A new and in-depth taxonomy is proposed to look deep into the existing methods. And in the end, we discuss some key issues and provide several directions for further researches. We believe that this work can not only give a clear overview of current researches, but encourage more opinions and solutions for KG quality management.

References

[1] J. Lehmann *et al.*, “DBpedia–A large-scale, multilingual knowledge base extracted from wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
[2] T. P. Tanon, G. Weikum, and F. Suchanek, “YAGO 4: A reasonable knowledge base,” in *Proc. Eur. Semantic Web Conf.*, 2020, pp. 583–596.
[3] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Commun. ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1306–1313.
[5] X. Dong *et al.*, “Knowledge vault: A web-scale approach to probabilistic knowledge fusion,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 601–610.
[6] L. Dietz, A. Kotov, and E. Meij, “Utilizing knowledge graphs for text-centric information retrieval,” in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Informat. Retrieval*, 2018, pp. 1387–1390.
[7] S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao, “Answering natural language questions by subgraph matching over knowledge graphs,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 824–837, May 2017.
[8] X. Huang, J. Zhang, D. Li, and P. Li, “Knowledge graph embedding based question answering,” in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 105–113.
[9] S. Zhou *et al.*, “Interactive recommender system via knowledge graph-enhanced reinforcement learning,” in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Informat. Retrieval*, 2020, pp. 179–188.
[10] Q. Guo *et al.*, “A survey on knowledge graph-based recommender systems,” *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 07, 2020, doi: 10.1109/TKDE.2020.3028705.
[11] C. Rudnik, T. Ehrhart, O. Ferret, D. Teyssou, R. Troncy, and X. Tannier, “Searching news articles using an event knowledge graph leveraged by wikidata,” in *Proc. Companion Proc. World Wide Web Conf.*, 2019, pp. 1232–1239.
[12] P. Ernst, C. Meng, A. Siu, and G. Weikum, “KnowLife: A knowledge graph for health and life sciences,” in *Proc. IEEE 30th Int. Conf. Data Eng.*, 2014, pp. 1254–1257.
[13] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, “Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago,” *Semantic Web*, vol. 9, no. 1, pp. 77–129, 2018.
[14] S. Madnick and R. Wang, “Introduction to total data quality management (TDQM) research program,” *Total Data Qual. Manage. Prog. MIT Sloan Sch. Manage*, vol. 1, 1992, Art. no. 92.
[15] S. E. Madnick, R. Y. Wang, Y. W. Lee, and H. Zhu, “Overview and framework for data and information quality research,” *J. Data Informat. Qual.*, vol. 1, no. 1, pp. 1–22, 2009.
[16] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, “Quality assessment for linked data: A survey,” *Semantic Web*, vol. 7, no. 1, pp. 63–93, 2016.
[17] A. Bronselaer, R. De Mol, and G. De Tré, “A measure-theoretic foundation for data quality,” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 627–639, Apr. 2017.
[18] L. Ehrlinger, E. Rusz, and W. Wöß, “A survey of data quality measurement and monitoring tools,” 2019, *arXiv:1907.08138*.
[19] Z. Abedjan *et al.*, “Detecting data errors: Where are we and what needs to be done?,” *Proc. VLDB Endowment*, vol. 9, no. 12, pp. 993–1004, 2016.
[20] S. Sadiq, N. K. Yeganeh, and M. Indulska, “20 years of data quality research: Themes, trends and synergies,” in *Proc. 22nd Australas. Database Conf.*, 2011, pp. 153–162.
[21] C. Batini, A. Rula, M. Scannapieco, and G. Viscusi, “From data quality to big data quality,” *J. Database Manage.*, vol. 26, no. 1, pp. 60–82, 2015.
[22] L. Cai and Y. Zhu, “The challenges of data quality and data quality assessment in the big data era,” *Data Sci. J.*, vol. 14, p. 2, 2015.
[23] I. Taleb, M. A. Serhani, and R. Dssouli, “Big data quality: A survey,” in *Proc. IEEE Int. Congr. Big Data Congr.*, 2018, pp. 166–173.
[24] N. Drummond and R. Shearer, “The open world assumption,” in *Proc. eSI Workshop, Closed World Databases Meets Open World Semantic Web*, p. 1, 2006.
[25] S. Tiwari, D. Gaurav, A. Srivastava, C. Rai, and K. Abhishek, “A preliminary study of knowledge graphs and their construction,” in *Emerg. Technol. Data Mining Informat. Security: Proc. IEMIS 2020*, vol. 3. Berlin, Germany: Springer, 2021, pp. 11–20.
[26] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor, “Industry-scale knowledge graphs: Lessons and challenges,” *Commun. ACM*, vol. 62, no. 8, pp. 36–43, 2019.
[27] S. Tiwari, F. N. Al-Aswadi, and D. Gaurav, “Recent trends in knowledge graphs: Theory and practice,” *Soft Comput.*, vol. 25, no. 13, pp. 8337–8355, 2021.
[28] X. Chen, S. Jia, and Y. Xiang, “A review: Knowledge reasoning over knowledge graph,” *Expert Syst. Appl.*, vol. 141, 2020, Art. no. 112948.
[29] D. Stepanova, M. H. Gad-Elrab, and V. T. Ho, “Rule induction and reasoning over knowledge graphs,” in *Reasoning Web International Summer School*, Berlin, Germany: Springer, 2018, pp. 142–172.
[30] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” *Proc. IEEE*, vol. 104, no. 1, pp. 11–33, Jan. 2016.
[31] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.
[32] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.
[33] L. L. Pipino, Y. W. Lee, and R. Y. Wang, “Data quality assessment,” *Commun. ACM*, vol. 45, no. 4, pp. 211–218, 2002.
[34] W. Fan and F. Geerts, “Foundations of data quality management,” *Synth. Lectures Data Manage.*, vol. 4, no. 5, pp. 1–217, 2012.
[35] W. Fan, “Data quality: From theory to practice,” *ACM SIGMOD Rec.*, vol. 44, no. 3, pp. 7–18, 2015.
[36] M. Abdallah, “Big data quality challenges,” in *Proc. IEEE Int. Conf. Big Data Comput. Intell.*, 2019, pp. 1–3.
[37] J. Debattista, C. Lange, S. Auer, and D. Cortis, “Evaluating the quality of the LOD cloud: An empirical investigation,” *Semantic Web*, vol. 9, no. 6, pp. 859–901, 2018.
[38] A. Piscopo and E. Simperl, “What we talk about when we talk about wikidata quality: A literature survey,” in *Proc. 15th Int. Symp. Open Collaboration*, 2019, pp. 1–11.
[39] S. Issa, O. Adekunle, F. Hamdi, S. S.-S. Cherfi, M. Dumontier, and A. Zaveri, “Knowledge graph completeness: A systematic literature review,” *IEEE Access*, vol. 9, pp. 31 322–31 339, 2021.
[40] F. Panse and F. Naumann, “Evaluation of duplicate detection algorithms: From quality measures to test data generation,” in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 2373–2376.
[41] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” *Semantic Web*, vol. 8, no. 3, pp. 489–508, 2017.
[42] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, “A survey on knowledge graphs: Representation, acquisition, and applications,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.
[43] R. Y. Wang and D. M. Strong, “Beyond accuracy: What data quality means to data consumers,” *J. Manage. Informat. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.
[44] Y. Wand and R. Y. Wang, “Anchoring data quality dimensions in ontological foundations,” *Commun. ACM*, vol. 39, no. 11, pp. 86–95, 1996.
[45] T. C. Redman, *Data Quality: The Field Guide*. Bedford, MA, USA: Digital, 2001.
[46] H. Chen, G. Cao, J. Chen, and J. Ding, “A practical framework for evaluating the quality of knowledge graph,” in *Proc. China Conf. Knowl. Graph Semantic Comput.*, 2019, pp. 111–122.
[47] V. Jayawardene, S. Sadiq, and M. Indulska, “The curse of dimensionality in data quality,” in *Proc. Asian Conf. Inf. Syst.*, 2013, p. 165.
[48] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, “AMIE: Association rule mining under incomplete evidence in ontological knowledge bases,” in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 413–422.
[49] B. K. Kahn, D. M. Strong, and R. Y. Wang, “Information quality benchmarks: Product and service performance,” *Commun. ACM*, vol. 45, no. 4, pp. 184–192, 2002.
[50] S. Sadiq *et al.*, “Data quality: The role of empiricism,” *ACM SIGMOD Rec.*, vol. 46, no. 4, pp. 35–43, 2018.
[51] H. V. Jagadish *et al.*, “Big data and its technical challenges,” *Commun. ACM*, vol. 57, no. 7, pp. 86–94, 2014.
[52] X. L. Dong and D. Srivastava, “Knowledge curation and knowledge fusion: Challenges, models and applications,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 2063–2066.
[53] P. Ojha and P. Talukdar, “Kgeval: Accuracy estimation of automatically constructed knowledge graphs,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1741–1750.
[54] T. Mitchell *et al.*, “Never-ending learning,” *Commun. ACM*, vol. 61, no. 5, pp. 103–115, 2018.
[55] N. Lao, T. Mitchell, and W. Cohen, “Random walk inference and learning in a large scale knowledge base,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 529–539.

[56] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang, "Efficient knowledge graph accuracy evaluation," 2019, *arXiv:1907.09657*.

[57] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.

[58] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?," *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[59] A. Zaveri *et al.*, "User-driven quality evaluation of dbpedia," in *Proc. 9th Int. Conf. Semantic Syst.*, 2013, pp. 97–104.

[60] D. Kontokostas, A. Zaveri, S. Auer, and J. Lehmann, "Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data," in *Proc. Int. Conf. Knowl. Eng. Semantic Web*, 2013, pp. 265–272.

[61] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann, "Crowdsourcing linked data quality assessment," in *Proc. Int. Semantic Web Conf.*, 2013, pp. 260–276.

[62] 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*, vol. 9, no. 3, pp. 303–335, 2018.

[63] L. Jiang, L. Chen, and Z. Chen, "Knowledge base enhancement via data facts and crowdsourcing," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 1109–1119.

[64] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.

[65] H. Paulheim and C. Bizer, "Improving the quality of linked data using statistical distributions," *Int. J. Semantic Web Informat. Syst.*, vol. 10, no. 2, pp. 63–86, 2014.

[66] D. Wienand and H. Paulheim, "Detecting incorrect numerical data in dbpedia," in *Proc. Eur. Semantic Web Conf.*, 2014, pp. 504–518.

[67] D. Fleischhacker, H. Paulheim, V. Bryl, J. Völker, and C. Bizer, "Detecting errors in numerical linked data using cross-checked outlier detection," in *Proc. Int. Semantic Web Conf.*, 2014, pp. 357–372.

[68] H. Paulheim, "Identifying wrong links between datasets by multi-dimensional outlier detection," in *Proc. 3rd Int. Workshop Debugging Ontologies Ontology Mappings*, 2014, pp. 27–38.

[69] H. Paulheim and H. Stuckenschmidt, "Fast approximate a-box consistency checking using machine learning," in *Proc. Eur. Semantic Web Conf.*, 2016, pp. 135–150.

[70] U. Lösch, S. Bloehdorn, and A. Rettinger, "Graph kernels for RDF data," in *Proc. Extended Semantic Web Conf.*, 2012, pp. 134–148.

[71] H. Paulheim and R. Meusel, "A decomposition of the outlier detection problem into a set of supervised learning problems," *Mach. Learn.*, vol. 100, no. 2, pp. 509–531, 2015.

[72] A. Melo and H. Paulheim, "Detection of relation assertion errors in knowledge graphs," in *Proc. Knowl. Capture Conf.*, 2017, pp. 1–8.

[73] A. Melo and H. Paulheim, "Automatic detection of relation assertion errors and induction of relation constraints," *Semantic Web*, vol. 11, no. 5, pp. 801–830, 2020.

[74] D. Gerber *et al.*, "Defacto—temporal and multilingual deep fact validation," *J. Web Semantics*, vol. 35, pp. 85–101, 2015.

[75] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, "Tracy: Tracing facts over knowledge graphs and text," in *Proc. World Wide Web Conf.*, 2019, pp. 3516–3520.

[76] F. Li, X. L. Dong, A. Langen, and Y. Li, "Knowledge verification for long-tail verticals," *Proc. VLDB Endowment*, vol. 10, no. 11, pp. 1370–1381, 2017.

[77] Y. Li *et al.*, "A survey on truth discovery," *ACM Sigkdd Explorations Newslett.*, vol. 17, no. 2, pp. 1–16, 2016.

[78] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: Is the problem solved?," 2015, *arXiv:1503.00303*.

[79] S. Liu, M. d'Aquin, and E. Motta, "Measuring accuracy of triples in knowledge graphs," in *Proc. Int. Conf. Lang. Data Knowl.*, 2017, pp. 343–357.

[80] B. Schäfer, P. Ristoski, and H. Paulheim, "What is special about bethlehem, pennsylvania? identifying unexpected facts about dbpedia entities," in *Proc. CEUR Workshop Proc.*, 2015, pp. Paper–46.

[81] J. Liang, Y. Xiao, Y. Zhang, S.-W. Hwang, and H. Wang, "Graph-based wrong IsA relation detection in a large-scale lexical taxonomy," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1178–1184.

[82] G. Zhang and C. Li, "Maverick: A system for discovering exceptional facts from knowledge graphs," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 1934–1937, 2018.

[83] S. Jia, Y. Xiang, X. Chen, and K. Wang, "Triple trustworthiness measurement for knowledge graph," in *Proc. World Wide Web Conf.*, 2019, pp. 2865–2871.

[84] X. Wang, X. L. Dong, and A. Meliou, "Data X-ray: A diagnostic tool for data errors," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1231–1245.

[85] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge base completion via search-based question answering," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 515–526.

[86] H. Paulheim and C. Bizer, "Type inference on noisy RDF data," in *Proc. Int. Semantic Web Conf.*, 2013, pp. 510–525.

[87] A. Melo and H. Paulheim, "An approach to correction of erroneous links in knowledge graphs," in *Proc. CEUR Workshop Proc.*, 2017, pp. 54–57.

[88] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl. Based Syst.*, vol. 151, pp. 78–94, 2018.

[89] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, pp. 1–49, 2021.

[90] R. Xie *et al.*, "Representation learning of knowledge graphs with hierarchical types," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2965–2971.

[91] Z. Wang, J. Li, Z. Liu, and J. Tang, "Text-enhanced representation learning for knowledge graph," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 4–17.

[92] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 192–202.

[93] P. Rosso, D. Yang, and P. Cudré-Mauroux, "Beyond triplets: Hyper-relational knowledge graph embedding for link prediction," in *Proc. Web Conf.*, 2020, pp. 1885–1896.

[94] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2659–2665.

[95] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier, "Connecting language and knowledge bases with embedding models for relation extraction," 2013, *arXiv:1307.7973*.

[96] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Proc. Adv. Neural Informat. Process. Syst.*, 2013, pp. 2787–2795.

[97] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.

[98] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.

[99] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.

[100] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 13tn AAAI Conf. Artif. Intell.*, 2016, pp. 985–991.

[101] M. Fan, Q. Zhou, E. Chang, and F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in *Proc. 28th Pacific Asia Conf. Lang. Informat. Comput.*, 2014, pp. 328–337.

[102] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 96–104.

[103] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.

[104] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 809–816.

[105] B. Yang, W.-T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," 2014, *arXiv:1412.6575*.

[106] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," 2018, *arXiv:1802.04868*.
[107] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, no. 1/4, pp. 164–189, 1927.
[108] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1955–1961.
[109] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[110] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Informat. Process. Syst.*, 2017, pp. 1025–1035.
[111] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[112] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, 2020, pp. 2704–2710.
[113] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.
[114] M. Simonovsky and N. Komodakis, "GraphVAE: Towards generation of small graphs using variational autoencoders," in *Int. Conf. Artif. Neural Netw.*, 2018, pp. 412–422.
[115] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.
[116] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5308–5317.
[117] W. Hu *et al.*, "Strategies for pre-training graph neural networks," 2019, *arXiv:1905.12265*.
[118] X. Liu *et al.*, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 12, 2021, doi: 10.1109/TKDE.2021.3119326.
[119] G. A. Gesese, R. Biswas, and H. Sack, "A comprehensive survey of knowledge graph embeddings with literals: Techniques and applications," in *Proc. DL4KG@ ESWC*, 2019, pp. 31–40.
[120] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[121] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[122] Ö. Sevgili, A. Panchenko, and C. Biemann, "Improving neural entity disambiguation with graph embeddings," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Student Res. Workshop*, 2019, pp. 315–322.
[123] J. Pujara, E. Augustine, and L. Getoor, "Sparsity and noise: Where knowledge graph embeddings fall short," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1751–1756.
[124] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li, "Realistic Re-evaluation of knowledge graph completion methods: An experimental study," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2020, pp. 1995–2010.
[125] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek, "Fast rule mining in ontological knowledge bases with AMIE+," *Very Large Data Bases J.*, vol. 24, no. 6, pp. 707–730, 2015.
[126] J. Lajus, L. Galárraga, and F. Suchanek, "Fast and exact rule mining with AMIE 3," in *Proc. Eur. Semantic Web Conf.*, 2020, pp. 36–52.
[127] Z. Wang and J. Li, "RDF2Rules: Learning rules from RDF knowledge bases by mining frequent predicate cycles," 2015, *arXiv:1512.07734*.
[128] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, "Exception-enriched rule learning from knowledge graphs," in *Proc. Int. Semantic Web Conf.*, 2016, pp. 234–251.
[129] S. Ortona, V. V. Meduri, and P. Papotti, "Robust discovery of positive and negative rules in knowledge bases," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 1168–1179.
[130] S. Ortona, V. V. Meduri, and P. Papotti, "Rudik: Rule discovery in knowledge bases," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 1946–1949, 2018.
[131] L. Galárraga, S. Razniewski, A. Amarilli, and F. M. Suchanek, "Predicting completeness in knowledge bases," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 375–383.
[132] C. Meilicke, M. W. Chekol, D. Ruffinelli, and H. Stuckenschmidt, "Anytime bottom-up rule learning for knowledge graph completion," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3137–3143.
[133] P. G. Omran, K. Wang, and Z. Wang, "Scalable rule learning via learning representation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2149–2155.
[134] V. T. Ho, D. Stepanova, M. H. Gad-Elrab, E. Kharlamov, and G. Weikum, "Rule learning from knowledge graphs guided by embedding models," in *Proc. Int. Semantic Web Conf.*, 2018, pp. 72–90.
[135] A. Sadeghian, M. Armandpour, P. Ding, and D. Z. Wang, "Drum: End-to-end differentiable rule mining on knowledge graphs," 2019, *arXiv:1911.00055*.
[136] N. Ahmadi, T.-T.-D. Truong, L.-H.-M. Dao, S. Ortona, and P. Papotti, "RuleHub: A public corpus of rules for knowledge graphs," *J. Data Informat. Qual.*, vol. 12, no. 4, pp. 1–22, 2020.
[137] J. Fan and G. Li, "Human-in-the-loop rule learning for data integration," *IEEE Data Eng. Bull.*, vol. 41, no. 2, pp. 104–115, 2018.
[138] B. De Meester, P. Heyvaert, D. Arndt, A. Dimou, and R. Verborgh, "RDF graph validation using rule-based reasoning," *Semantic Web*, vol. 12, no. 1, pp. 117–142, 2021.
[139] H. Paulheim and A. Gangemi, "Serving dbpedia with dolce–more than just adding a cherry on top," in *Proc. Int. Semantic Web Conf.*, 2015, pp. 180–196.
[140] H. Paulheim, "Data-driven joint debugging of the dbpedia mappings and ontology," in *Proc. Eur. Semantic Web Conf.*, 2017, pp. 404–418.
[141] T. Pellissier Tanon, C. Bourgaux, and F. Suchanek, "Learning how to correct a knowledge base from the edit history," in *Proc. World Wide Web Conf.*, 2019, pp. 1465–1475.
[142] C. Fürber and M. Hepp, "Using SPARQL and spin for data quality management on the Semantic Web," in *Proc. Int. Conf. Bus. Informat. Syst.*, 2010, pp. 35–46.
[143] D. Kontokostas *et al.*, "Test-driven evaluation of linked data quality," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 747–758.
[144] W. Fan, Y. Wu, and J. Xu, "Functional dependencies for graphs," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 1843–1857.
[145] W. Fan, X. Liu, and Y. Cao, "Parallel reasoning of graph functional dependencies," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 593–604.
[146] W. Fan, C. Hu, X. Liu, and P. Lu, "Discovering graph functional dependencies," *ACM Trans. Database Syst.*, vol. 45, no. 3, pp. 1–42, 2020.
[147] W. Fan, Z. Fan, C. Tian, and X. L. Dong, "Keys for graphs," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1590–1601, 2015.
[148] W. Fan and P. Lu, "Dependencies for graphs," *ACM Trans. Database Syst.*, vol. 44, no. 2, pp. 1–40, 2019.
[149] W. Fan, X. Liu, P. Lu, and C. Tian, "Catching numeric inconsistencies in graphs," *ACM Trans. Database Syst.*, vol. 45, no. 2, pp. 1–47, 2020.
[150] W. Fan, P. Lu, C. Tian, and J. Zhou, "Deducing certain fixes to graphs," *Proc. VLDB Endowment*, vol. 12, no. 7, pp. 752–765, 2019.
[151] S. Kwashie, L. Liu, J. Liu, M. Stumptner, J. Li, and L. Yang, "Certus: An effective entity resolution approach with graph differential dependencies (GDDs)," *Proc. VLDB Endowment*, vol. 12, no. 6, pp. 653–666, 2019.
[152] M. Alipourlangouri, "Temporal dependencies for graphs," in *Proc. Int. Conf. Manage. Data*, 2021, pp. 2881–2883.
[153] W. Fan, X. Wang, Y. Wu, and J. Xu, "Association rules with graph patterns," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1502–1513, 2015.
[154] W. Fan, R. Jin, M. Liu, P. Lu, C. Tian, and J. Zhou, "Capturing associations in graphs," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 1863–1876, 2020.
[155] M. H. Namaki, Y. Wu, Q. Song, P. Lin, and T. Ge, "Discovering graph temporal association rules," in *Proc. ACM Conf. Informat. Knowl. Manage.*, 2017, pp. 1697–1706.
[156] S. Song, H. Cheng, J. X. Yu, and L. Chen, "Repairing vertex labels under neighborhood constraints," *Proc. VLDB Endowment*, vol. 7, no. 11, pp. 987–998, 2014.
[157] S. Song, B. Liu, H. Cheng, J. X. Yu, and L. Chen, "Graph repairing under neighborhood constraints," *VLDB J.*, vol. 26, no. 5, pp. 611–635, 2017.
[158] Y. Cheng, L. Chen, Y. Yuan, and G. Wang, "Rule-based graph repairing: Semantic and efficient repairing methods," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 773–784.
[159] Y. Cheng, L. Chen, Y. Yuan, G. Wang, B. Li, and F. Jin, "Strict and flexible rule-based graph repairing," *IEEE Trans. Knowl. Data Eng.*, early access, Aug. 27, 2020, doi: 10.1109/TKDE.2020.3019817.

[160] C. Belth, X. Zheng, J. Vreeken, and D. Koutra, "What is normal, what is strange, and what is missing in a knowledge graph: Unified characterization via inductive summarization," in *Proc. Web Conf.*, 2020, pp. 1115–1126.
[161] J. Lajus and F. M. Suchanek, "Are all people married? determining obligatory attributes in knowledge bases," in *Proc. World Wide Web Conf.*, 2018, pp. 1115–1124.
[162] S. Kruse, A. Jentzsch, T. Papenbrock, Z. Kaoudi, J.-A. Quian é-Ruiz, and F. Naumann, "RDFind: Scalable conditional inclusion dependency discovery in rdf datasets," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 953–967.
[163] T.-K. Tran, M. H. Gad-Elrab, D. Stepanova, E. Kharlamov, and J. Strötgen, "Fast computation of explanations for inconsistency in large-scale knowledge graphs," in *Proc. Web Conf.*, 2020, pp. 2613–2619.
[164] Y. Chen and D. Z. Wang, "Knowledge expansion over probabilistic knowledge bases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 649–660.
[165] M. Chekol, G. Pirrò, J. Schoenfisch, and H. Stuckenschmidt, "Marrying uncertainty and time in knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 88–94.
[166] M. W. Chekol, G. Pirro, J. Schoenfisch, and H. Stuckenschmidt, "TeCoRe: Temporal conflict resolution in knowledge graphs," *Proc. VLDB Endowment*, vol. 10, pp. 1929–1932, 2017.
[167] S. Muggleton and L. De Raedt, "Inductive logic programming: Theory and methods," *J. Log. Prog.*, vol. 19, pp. 629–679, 1994.
[168] L. De Raedt, *Logical and Relational Learning*. Berlin, Germany: Springer, 2008.
[169] M. D. Tran, C. d'Amato, B. T. Nguyen, and A. G. Tettamanzi, "Comparing rule evaluation metrics for the evolutionary discovery of multi-relational association rules in the Semantic Web," in *Proc. Eur. Conf. Genet. Prog.*, 2018, pp. 289–305.
[170] T. P. Tanon, D. Stepanova, S. Razniewski, P. Mirza, and G. Weikum, "Completeness-aware rule learning from knowledge graphs," in *Proc. Int. Semantic Web Conf.*, 2017, pp. 507–525.
[171] K. Zupanc and J. Davis, "Estimating rule quality for knowledge base completion with the relationship between coverage assumption," in *Proc. World Wide Web Conf.*, 2018, pp. 1073–1081.
[172] M. Loster, D. Mottin, P. Papotti, J. Ehmüller, B. Feldmann, and F. Naumann, "Few-shot knowledge validation using rules," in *Proc. Conf.*, 2021, pp. 3314–3324.
[173] Q. Zeng, J. M. Patel, and D. Page, "QuickFOIL: Scalable inductive logic programming," *Proc. VLDB Endowment*, vol. 8, no. 3, pp. 197–208, 2014.
[174] S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld, "Learning first-order horn clauses from web text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 1088–1098.
[175] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
[176] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable learning of logical rules for knowledge base reasoning," 2017, *arXiv:1702.08367*.
[177] D. Arndt, B. De Meester, A. Dimou, R. Verborgh, and E. Mannens, "Using rule-based reasoning for RDF validation," in *Proc. Int. Joint Conf. Rules Reasoning*, 2017, pp. 22–36.
[178] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of SPARQL," *ACM Trans. Database Syst.*, vol. 34, no. 3, pp. 1–45, 2009.
[179] D. L. McGuinness *et al.*, "Owl web ontology language overview," *W3C Recommendation*, vol. 10, no. 10, 2004, Art. no. 2004.
[180] H. Knublauch and D. Kontokostas, "Shapes constraint language (SHACL)," *W3C Candidate Recommendation*, vol. 11, no. 8, p. 1, 2017.
[181] E. Prud'hommeaux, J. E. Labra Gayo, and H. Solbrig, "Shape expressions: An RDF validation and transformation language," in *Proc. 10th Int. Conf. Semantic Syst.*, 2014, pp. 32–40.
[182] T. Bosch, E. Acar, A. Nolle, and K. Eckert, "The role of reasoning for RDF validation," in *Proc. 11th Int. Conf. Semantic Syst.*, 2015, pp. 33–40.
[183] D. Tomaszuk, "Rdf validation: A brief survey," in *Proc. Int. Conf., Beyond Databases Architectures Struct.*, 2017, pp. 344–355.
[184] L. Bühmann, J. Lehmann, and P. Westphal, "Dl-learner—a framework for inductive learning on the Semantic Web," *J. Web Semantics*, vol. 39, pp. 15–24, 2016.
[185] C. Chai and G. Li, "Human-in-the-loop techniques in machine learning," *Data Eng.*, vol. 37, p. 16, 2020, Art. no. 37.
[186] Z. Chen, Q. Chen, and Z. Li, "A human-and-machine cooperative framework for entity resolution with quality guarantees," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 1405–1406.
[187] Z. Chen *et al.*, "Enabling quality control for entity resolution: A human and machine cooperation framework," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 1156–1167.
[188] C. Chai, L. Cao, G. Li, J. Li, Y. Luo, and S. Madden, "Human-in-the-loop outlier detection," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2020, pp. 19–33.
[189] P. Yao and D. Barbosa, "Typing errors in factual knowledge graphs: Severity and possible ways out," in *Proc. Web Conf.*, 2021, pp. 3305–3313.
[190] A. Arioua and A. Bonifati, "User-guided repairing of inconsistent knowledge bases," in *Proc. EDBT, Extending Database Technol.*, 2018, pp. 133–144.
[191] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, and H. Stuckenschmidt, "Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion," in *Proc. Int. Semantic Web Conf.*, 2018, pp. 3–20.
[192] J. Chen, X. Chen, I. Horrocks, E. B. Myklebust, and E. Jimenez-Ruiz, "Correcting knowledge base assertions," in *Proc. Web Conf.*, 2020, pp. 1537–1547.
[193] W. Zhang *et al.*, "Iteratively learning embeddings and rules for knowledge graph reasoning," in *Proc. World Wide Web Conf.*, 2019, pp. 2366–2377.
[194] S. Hao, C. Chai, G. Li, N. Tang, N. Wang, and X. Yu, "Outdated fact detection in knowledge bases," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 1890–1893.
[195] F. Zhang, Z. Li, D. Peng, and J. Cheng, "RDF for temporal data management–a survey," *Earth Sci. Inform.*, vol. 14, no. 2, pp. 563–599, 2021.

Bingcong Xue received the BS degree in computer science in 2020 from Peking University, Beijing, China, where she is currently working toward the master's degree with the Academy for Advanced Interdisciplinary Studies, Peking University. Her research interests include knowledge graph and natural language processing.

Lei Zou received the BS and PhD degrees in computer science from the Huazhong University of Science and Technology, in 2003 and 2009, respectively. He is currently a professor with the Wangxuan Institute of Computer Technology, Peking University. He is also a faculty member with Big Data Center, Peking University. His research interests include graph database and knowledge graph.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.