SEKA: Seeking Knowledge Graph Anomalies

Asara Senaratne
asara.senaratne@anu.edu.au
School of Computing, The Australian National University, Canberra, Australia
Supervised by: Prof. Peter Christen, Prof. Graham Williams, Dr. Pouya Ghiasnezhad Omran

ABSTRACT

Knowledge Graphs (KGs) form the backbone of many knowledge dependent applications such as search engines and digital personal assistants. KGs are generally constructed either manually or automatically using a variety of extraction techniques applied over multiple data sources. Due to the diverse quality of these data sources, there are likely anomalies introduced into any KG. Hence, it is unrealistic to expect a perfect archive of knowledge. Given how large KGs can be, manual validation is impractical, necessitating an automated approach for anomaly detection in KGs. To improve KG quality, and to identify interesting and abnormal triples (edges) and entities (nodes) that are worth investigating, we introduce SEKA, a novel unsupervised approach to detect anomalous triples and entities in a KG using both the structural characteristics and the content of edges and nodes of the graph. While an anomaly can be an interesting or unusual discovery, such as a fraudulent transaction requiring human intervention, anomaly detection can also identify potential errors. We propose a novel approach named Corroborative Path Algorithm to generate a matrix of semantic features, which we then use to train a one-class Support Vector Machine to identify abnormal triples and entities with no dependency on external sources. We evaluate our approach on four real-world KGs demonstrating the ability of SEKA to detect anomalies, and to outperform comparative baselines.

KEYWORDS

Knowledge graph quality enhancement, semantic features, unsupervised anomaly detection, one-class classifier

ACM Reference Format:
Asara Senaratne. 2023. SEKA: Seeking Knowledge Graph Anomalies. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543873.3587536

1 INTRODUCTION

Intelligent assistants, such as Alexa and Siri, introduced Artificial Intelligence based communication agents to millions of households worldwide. An agent requires knowledge to give a meaningful and logical answer to a question posted by a human. Knowledge representation and reasoning, inspired by human problem solving, represent knowledge for intelligent agents to gain the ability to solve complex tasks. Hence, at present, Knowledge Graphs (KG) as a form of structured human knowledge representation have drawn research attention both from industry and academia [6].

When constructing a KG, it can either be manually curated by experts, manually generated by volunteers, automatically extracted from text via hand-crafted or learned rules, or automatically extracted from unstructured text using machine learning techniques. Irrespective of the approach, the presence of anomalies is inevitable as there is no perfect source of data [4].

While validation techniques such as Shapes Constraint Language (SHACL) and Shape Expressions (ShEx) offer insights into the structure of a KG [15], not every real-world KG has a shape-based layer to facilitate such validation. Furthermore, these techniques propose what should be in a KG as opposed to what should not be in the KG. Similarly, rule-based reasoners and constraints engines for KG validation [7] only find common patterns of errors. Even though errors can be represented via pre-known patterns, an anomaly cannot be guessed before being detected. While every error can be considered as an anomaly in data, not every anomaly is an error. Non-erroneous anomalies have the potential to uncover interesting information, thus discovering new knowledge from a KG. Although there exist other approaches to detect anomalies in KGs, they are either domain-specific [26], require human involvement [8], or are dependent on external resources [7].

We propose SEKA, an unsupervised anomaly detection approach to identify anomalous triples and entities in a KG using both the structural properties and content of the graph. With the motivation received from our recent work in anomaly detection [20, 21], our aim is to discover abnormal triples and entities that provide interesting, unusual, contradicting, semantically incorrect, redundant, invalid, incomplete, and missing information in KGs, as provided with examples in Table 1. With the introduction of SEKA, our main contribution is the improvement of data quality in KGs, thereby improving the reliability of applications using KGs as the backbone. Furthermore, we also contribute to the area of KG enhancement with the introduction of the Corroborative Path Algorithm (CPA), an algorithm dedicated for anomaly detection in KGs.

2 PROBLEM DEFINITION

Much of today's data can be represented as graphs, ranging from social networks to bibliographic citations. Nodes in such graphs generally represent real-world entities, while edges represent relationships between them. Both nodes and edges in a graph can have attributes that characterize the entities and their relationships. Relationships are either explicitly known (such as friendships in a social network or citations in bibliographic data), or they can be inferred using record linkage or link prediction algorithms (such as two babies are siblings because they have the same mother) [20].

WWW '23 Companion, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04.
https://doi.org/10.1145/3543873.3587536

Table 1: Some examples of interesting anomalies detected by SEKA.

Anomalous Triple	Anomaly Explained
<DonaldTrump, marriedTo, MarlaMaples> <MarlaMaples, hasChild, DonaldTrump>	Two contradicting relationships between the same two people, while one triple is wrong.
<EthelricArchbishopOfYork><hasSuccessor><AelfricPuttoc> <EthelricArchbishopOfYork><hasPredecessor><AelfricPuttoc>	It is unusual for one person to be both predecessor and successor of another person. However, this can be a possibility in politics and religion.
<Karl-HermannKnoblauch, hasWonPrize, KingdomOfPrussia>	The predicate "hasWonPrize" is generally followed by the name of the prize won instead of the object for which it was awarded, thus making the predicate usage ambiguous.
<Ain'tTooProudToBeg, isOfGenre, rock> <Ain'tTooProudToBeg, isOfGenre, music> <Ain'tTooProudToBeg, isOfGenre, popularMusic>	The range of this predicate is not well defined. Hence, the same subject and predicate have different objects making the predicate usage ambiguous.
<AMGrapper><produced><BettaHaveMoney> <AMGrapper><produced><BettaHaveMoney2001>	These two facts seem to provide redundant information causing entity redundancy.
<Aristotle, bornOn, "348-##-##">	The object contains a partial date making the literal value invalid.
<Marcelona, bornIn, Mozambique> <Marcelona, hasSuccessor, Mozambique>	Entity "Mozambique" is treated both as a person and location causing entity ambiguity.
<9thWonder, produced, TheDreamMerchantVol2>	The subject is missing its corresponding "created" triple which other triples related to music albums have, thus making the triple sparse.
<Neuromance, isOfGenre, hacker>	"Hacker" is not a common type of genre. Hence the object seems abnormal.
<person/11203, hasName, A.>	"A." alone cannot be the name of a person, thus creating an abnormal object
<SQL, hasDefinition, "">	A triple with a missing literal value.
<dataset/411>	This is an anomalous entity (in DSKG) as it is the only dataset with eleven creators, whereas other datasets have at most five creators.

Any graph representing real-world data likely contains both nodes and edges that are abnormal or unusual, and identifying these can be important for outlier detection in applications such as crime and fraud detection, viral marketing, or to identify subsets of nodes and edges that are useful in active learning for manual classification. At present, there exist various methods to detect anomalous nodes and edges based on the characteristics of the underlying structure of a graph, such as the density of the neighborhood of a node, the number of outgoing and incoming edges, to name a few. Also, identifying graph abnormalities based on temporal aspects is an active field of study [19].

While existing graph anomaly detection techniques focus on structural properties of a KG to identify anomalies, to the best of our knowledge there are no approaches in the literature that aim to identify both abnormal nodes and edges via the data associated with them. Furthermore, existing techniques prioritize error detection over anomaly detection. Hence, detecting anomalies solely based on the attribute values associated with the nodes and edges of a KG remains less explored.

Due to the increasing availability of large KGs, we believe it is important to develop mechanisms to reveal anomalies in an unsupervised manner, utilizing the attributes that are available in a KG. The essence of attribute-based graph anomaly detection is the knowledge that cannot be identified from the structure of a graph alone, but instead can be discovered by analyzing the data associated with the nodes and edges of an attributed graph.

Definition 2.1 (Attributed graph). We consider an attributed graph $G = (V, E, ^{\circ}, L)$, where V is the set of vertices and $v \in V$ is a vertex in G; $E \subseteq V \times V$ is the set of edges and $e = (u, v) \in E$ is an edge between two vertices u and v; $^{\circ}$ is the domain of attributes; and L is the function that assigns attributes to vertices and edges, where L can be any automated or manual means of generating the attributes. We use $l(v)$ and $l(e)$ to represent the attributes of vertex v and edge e, respectively [24].

Considering an edge-labelled graph which is a type of attributed graph with a single categorical attribute (label) for the edge [24], we define a *path* in the graph as follows:

Definition 2.2 (Path). In an edge-labelled graph G, a path P is defined as a directed, labelled sequence of vertices and edges $v_1 \xrightarrow{p_1} v_2 \xrightarrow{p_2} \ldots \xrightarrow{p_{k-1}} v_k$ in G, where $v_i \in V$ denotes real-world entities, p_i represents the predicate (edge label) of the directed edge that connects vertex i to $i+1$, and k denotes the length of the path.

Considering a directed edge-labelled graph as defined above, we define the *neighborhood* of a vertex as follows [24]:

Definition 2.3 (Neighborhood). For an edge-labelled graph G, the neighborhood $N_G(v)$ of a vertex $v \in V$ is its set of all neighbors of v, $N_G(v) = \{u | \{u, v\} \in E\}; u \in V$.

Considering a KG which is a directed edge-labelled graph [24], we now formally define the problem under study as follows:

Definition 2.4 (Anomaly detection in knowledge graphs). Given a knowledge graph G and a statement of fact $\mathcal{F} = (s, p, o)$, where $(s, o) \in V$ and $p \in E$, abnormal fact detection in G is the process of learning the relationship p using $L_e(P)$ of all the paths P between (s, o) with $0.5 \leq k \leq 2$ to classify the edge $s \xrightarrow{p} o$ in G into one of the two classes *normal* or *abnormal*. Similarly, abnormal entity (node) detection (say for s) in G is the process of learning $l(s)$ and $l(N_G(s))$ to determine the class of $s \in G$. An entity s or a fact $\mathcal{F}$ is classified *abnormal*, if the associated data deviates significantly from the rest of the data under consideration.

Simply put, we view the anomaly detection problem as an unsupervised learning task that validates a proposed fact $\mathcal{F}$ or node s by determining if the data associated with the fact and node is implied by the data within the KG. To ensure high data quality, and to extract accurate insights out of data, it is critical that such abnormalities are detected so they can be investigated.

3 STATE OF THE ART

Anomaly detection in KGs has received much attention as automated methods of constructing KGs prioritize data integrity [5]. There exist methods for error detection in KGs, where each approach may target specific types of information [17]. Some approaches take advantage of entity type information to perform clustering-based outlier detection [2], however entity types can either be absent or only be partially available in a KG. Another set of approaches use path ranking [13] or path-based rule mining for error checking [22]. While path ranking methods have limitations in the coverage due to the probabilistic approach adopted for path discovery, path-based rule mining methods can only discover regularities in graphs as opposed to discovering abnormalities.

Most recent studies propose employing supervised classifiers to evaluate every triple based on different features, including entity categories, path features, in/out-degrees, as well as embedding representations of entities and relations [13]. However, ground truth data may not be available to train such classifiers. Alternatively, there have been efforts to utilize external information sources [7, 14, 25], such as related web pages [11] and annotations [8, 12], to facilitate anomaly detection in KGs. While having external resources can be valuable for this task, acquiring such supplemental information can be time consuming and expensive. Furthermore, there are approaches that only aim to identify a single type of anomaly [13], approaches that are KG dependent [16, 26], methods requiring human intervention [8], and embedding methods that can only consider structural properties while eliminating the content associated with entities [1, 23, 27].

Even though there have been many different approaches proposed for error/anomaly detection in KGs, they fall short in providing a sound solution either because they only perform error detection, or they are introduced for one specific KG. To the best of our knowledge, there is no approach that can detect both anomalous triples and entities, detect a multitude of anomalies, domain and KG independent, scalable to large KGs, unsupervised, and supports anomaly generalization.

4 PROPOSED APPROACH

Our proposed approach (SEKA) performs fact anomaly detection considering the triples in a KG to identify anomalous triples, and entity anomaly detection to identify anomalous entities in a KG. Table 1 shows examples of some anomalies that can be detected by our approach.

To construct features that highlight the structural properties of a KG, we introduce a variation of the Path Rank Algorithm (PRA) [10], named the Corroborative Path Algorithm (CPA). While PRA is widely used for the task of link prediction [10], CPA addresses the task of anomaly detection by considering corroborative (alternative) paths between two entities in a fact (triple) bounded by length to construct binary features. We mark the existence or non-existence of a path between two entities with a binary *True/False* value to construct a binary feature vector for each triple in the KG, therefore forming a matrix of binary features. To reduce complexity and to improve performance, CPA adopts a depth-first search bounded by the maximum length of a path (default length is two) as opposed to random walks used in PRA.

For entity anomaly detection, we generate content-based binary features by considering data quality aspects of the literals such as the presence/absence, validity/invalidity, and so on, of a triple [21]. This way, we can identify abnormal entities in a KG considering both structure and content.

We then train a one-class Support Vector Machine (SVM) to perform unsupervised anomaly detection on the generated feature matrix [20]. Once we have identified anomalies, we can remove the identified abnormal triples from the KG and use the refined graph in a downstream task such as Knowledge Graph Completion (KGC), or we can perform manual validation with the involvement of domain experts.

The novelty of SEKA is that it (1) can consider structure and content to detect both anomalous triples and entities, (2) is domain independent, and (3) is independent of external resources. Furthermore, to the best of our knowledge, there is no other work in the domain of KG validation that can detect both anomalous facts and entities in a KG. The novelty of CPA is that it has lower complexity and substantially lower run times compared to traditional PRA, making CPA scalable and well suited for anomaly detection in large KGs. Furthermore, with the generation of semantic features, CPA has the capability of detecting semantic anomalies as we consider the sequence of occurrence of paths between two entities as the features, which has the potential to identify rare path occurrences such as *marriedTo—hasChild* as per an example from Table 1.

5 METHODOLOGY

SEKA can perform two anomaly detection tasks as shown in Figure 1. The first is fact anomaly detection which identifies anomalous triples using CPA, as shown by the first (top) matrix in Figure 1. In this matrix (Matrix I), a row represents a triple from the KG on left, the features are the alternative paths between entities with a path length of up to two. These features are binary and indicate the existence or non-existence of a path between two entities. For example, the two entities *John* and *Canada* in the triple *livesIn(John, Canada)* have the alternative paths *citizenOf* and *citizenOf—locatedIn* as indicated by the binary value 1 (true).

The second task is to identify anomalous entities, where we identify abnormal entities considering both structural properties and content associated with an entity, as shown by the second (bottom) matrix in Figure 1 (Matrix II). The aim of this task is to identify entities that can be anomalous even when there are no anomalous facts associated with them. For example, consider the node *Mary* in the KG of Figure 1, which is abnormal due to absence of associations compared to other nodes in the KG. In the second matrix, a row represents an entity from the KG on left, while the columns represent three types of features. The first set of features (structural) indicates the predicates the entity is associated with within its neighborhood. The second set of features (content-based) highlight data quality aspects by referring to the literal-based triples associated with an entity.

We obtain the third set of features (structural) via the disjunction of the feature vectors from Matrix I, where the entity is the subject of the triple. We can then either use Matrix-I or Matrix-II as input to a one-class SVM for unsupervised learning [20] to obtain abnormal triples or entities, respectively.

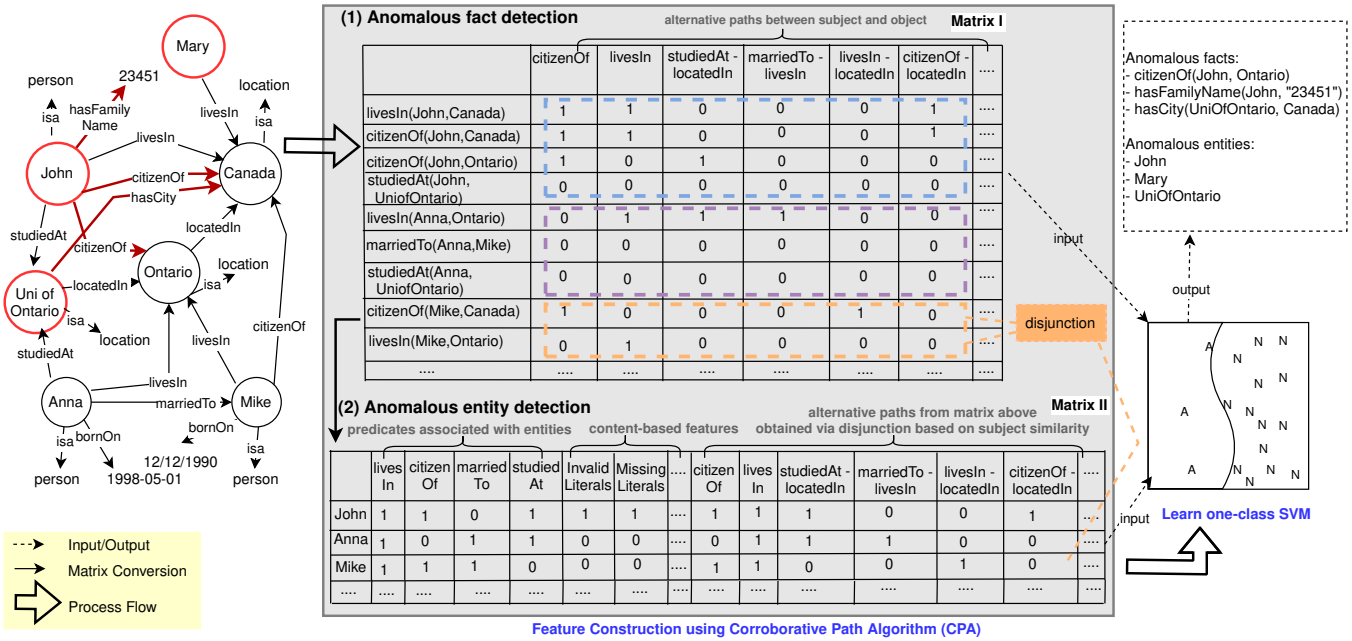

Figure 1: Overview of SEKA, the anomaly detection process to identify anomalous triples and entities in a KG, as described in Section 5. The abnormal triples and entities are marked in red in the KG on the left-hand side of the figure.

We used the four real-world KGs YAGO-1[1], KBpedia[2], Wikidata[3], and DSKG[4] to evaluate SEKA. We selected KGs in such a way that they are of different types, data qualities, sizes, and belong to different domains. As these KGs do not have labelled data, we manually corrupted existing triples by replacing either of subject, predicate, or object in a triple by another of the same type, or a different type (semantically different). We corrupted different percentages of triples (10%, 20% and 30%) to evaluate the suitability of SEKA for anomaly detection.

First, we conducted experiments using the three baselines of SEKA: (1) **PaTyBRED** [13] is used for the detection of relation assertion errors in KGs, which incorporates type and path features into local relation classifiers; (2) **SDValidate** [18] relies on statistical distributions of types and relations, and applies outlier detection to detect erroneous relation assertions; (3) **KGTtm** [9] synthesizes the internal semantic information in the triples and the global inference information of a KG to obtain the trustworthiness measurement.

Next, we evaluated the performance of CPA against PRA in terms of run time, precision, and recall. We consider PRA as the baseline of CPA, as CPA is a variation of PRA that is dedicated for anomaly detection, while PRA is widely used for link prediction [10]. We conducted both these experiments with synthetically generated anomalies. Our third experiment was to assess the performance of entity anomaly detection with synthetic anomalies, in terms of run time, precision, and recall. Next, we performed a manual evaluation on the four KGs without the introduction of synthetic anomalies, where we manually verified the top fifty (based on one-class SVM anomaly score [20]) abnormal triples in these KGs as identified by SEKA. We obtained the examples in Table 1 via this manual evaluation. This experiment further demonstrated the generalizability of SEKA, and its ability to detect any abnormalities not covered by the synthetically generated anomalies, as SEKA has no pre-defined rules or patterns defining anomalies.

6 RESULTS

We compared SEKA with the baselines PaTyBRED [13], SDValidate [18], and KGTtm [9] using precision and recall values to determine how well each approach performed in identifying the anomalies. As per Table 2 that shows the experimental results with 10% of the triples corrupted, SEKA performs better in comparison to all the baselines under consideration. In comparison to these baselines, SEKA achieves an increase of up to 12%, 15%, 16%, 17% in precision, and an increase of up to 15%, 16%, 18%, 14% in recall for YAGO-1, DSKG, Wikidata, and KBpedia, respectively. Hence, SEKA outperforms all baselines with its capability to identify anomalous triples. Similarly, SEKA outperformed the baselines with 20% and 30% triples corrupted.

To evaluate CPA versus PRA, we conducted experiments to assess the run time, and the quality of the anomalies detected by probabilistic feature generation versus binary feature generation. In Table 3, we show the experimental results obtained with 10% of the triples corrupted in the four KGs. As can be seen from Table 3, CPA outperforms PRA on all four KGs with substantially reduced run times, and higher precision and recall values. This demonstrates the suitability of CPA in generating features required for anomaly detection, compared to PRA which is dedicated for link prediction task. CPA achieves an increase of up to 8% and 10% in precision

[1]https://yago-knowledge.org/downloads/yago-1
[2]https://kbpedia.org/
[3]https://www.wikidata.org/wiki/Wikidata:Main_Page
[4]http://dskg.org/

Table 2: Comparison of SEKA with baselines with 10% of triples corrupted. The best results are shown in bold.

Approach	YAGO-1		DSKG		Wikidata		KBpedia	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
PaTyBRED	0.87	0.86	0.84	0.83	0.72	0.72	0.77	0.75
SDValidate	0.82	0.80	0.81	0.80	0.70	0.68	0.71	0.71
KGTtm	0.86	0.84	0.88	0.83	0.77	0.77	0.70	0.69
SEKA	**0.92**	**0.92**	**0.93**	**0.93**	**0.81**	**0.80**	**0.82**	**0.81**

Table 3: Comparison of performance between PRA and CPA for general fact anomaly detection with 10% of triples corrupted. The best results are shown in bold.

KG	Approach	Run time (min)	Precision	Recall
YAGO-1	PRA	178	0.85	0.84
	CPA	**121**	**0.92**	**0.92**
KBpedia	PRA	88	0.80	0.79
	CPA	**48**	**0.82**	**0.81**
Wikidata	PRA	301	0.79	0.78
	CPA	**232**	**0.81**	**0.80**
DSKG	PRA	137	0.92	0.90
	CPA	**72**	**0.93**	**0.93**

and recall, respectively, and a decrease in the run time of up to 47% in comparison with PRA. We obtained similar results with 20% and 30% of the triples corrupted in each of the four KGs.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented an approach to study the problem of discovering anomalous triples and entities in a Knowledge Graph (KG) in an unsupervised manner. Our approach can identify anomalies related to both structure and content of the KG, it is independent from external resources, and has the ability to identify a multitude of anomalies.

We conducted different experiments using four real-world KGs with synthetic anomalies introduced to demonstrate the state-of-the-art performance of our approach in anomaly detection. The results of our evaluation show how SEKA can consistently outperform its baselines.

As future work, we will be improving the feature pruning step of CPA. We also aim to generalize the identified anomalies with the use of a constraint-based language such as SHACL and Graph Functional Dependencies (GFD) [3], as we can use these validations during the process of KG construction to filter triples. Furthermore, we will develop a taxonomy of KG anomaly types so that the anomalies detected by SEKA can be automatically categorized in to one of the pre-defined categories for ease of correction.

REFERENCES

[1] Farhad Abedini, Mohammad Reza Keyvanpour, and Mohammad Bagher Menhaj. 2020. Correction Tower: A general embedding method of the error recognition for the knowledge graph correction. *IJPRAI* 34, 10 (2020), 2059034.

[2] Jeremy Debattista, Christoph Lange, and Sören Auer. 2016. A preliminary investigation towards improving linked data quality using distance-based outlier detection. In *JIST*. Springer, Cham, 116–124.

[3] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. 2020. Discovering graph functional dependencies. *ACM Transactions on Database Systems (TODS)* 45, 3 (2020), 1–42.

[4] Dieter Fensel, U Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. 2020. *Knowledge graphs*. Springer, Switzerland.

[5] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism detection in wikidata. In *CIKM*. ACM, New York, 327–336.

[6] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *CSUR* 54, 4 (2021), 1–37.

[7] Elwin Huaman, Amar Tauqeer, and Anna Fensel. 2021. Towards Knowledge Graphs Validation Through Weighted Knowledge Sources. In *KGSW*. Springer, Cham, 47–60.

[8] Manuela Nayantara Jeyaraj, Srinath Perera, Malith Jayasinghe, and Nadheesh Jihan. 2019. Probabilistic error detection model for knowledge graph refinement. In *CICLing (CiCLing'19)*. Springer, Switzerland.

[9] Shengbin Jia, Yang Xiang, Xiaojun Chen, and Kun Wang. 2019. Triple trustworthiness measurement for knowledge graph. In *WWW*. 2865–2871.

[10] Ni Lao and William W Cohen. 2010. Fast query execution for retrieval models based on path-constrained random walks. In *SIGKDD*. 881–888.

[11] Jens Lehmann, Daniel Gerber, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. 2012. Defacto-deep fact validation. In *International semantic web conference*. Springer, Berlin, 312–327.

[12] Yezi Liu. 2021. *Error Detection in Knowledge Graphs*. Ph. D. Dissertation. Graduate and Professional School of Texas A and M University.

[13] André Melo and Heiko Paulheim. 2017. Detection of relation assertion errors in knowledge graphs. In *Knowledge Capture Conference*. ACM, New York, 1–8.

[14] Satoshi Nakamura, Shinji Konishi, Adam Jatowt, Hiroaki Ohshima, Hiroyuki Kondo, Taro Tezuka, Satoshi Oyama, and Katsumi Tanaka. 2007. Trustworthiness analysis of web search results. In *TPDL*. Springer, Berlin, 38–49.

[15] Pouya Ghiasnezhad Omran, Kerry Taylor, Sergio Rodriguez Mendez, and Armin Haller. 2021. Learning SHACL Shapes from Knowledge Graphs. *Semantic Web* (2021).

[16] Heiko Paulheim. 2017. Data-driven joint debugging of the dbpedia mappings and ontology. In *ESWC*. Springer, cham, 404–418.

[17] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[18] Heiko Paulheim and Christian Bizer. 2014. Improving the quality of linked data using statistical distributions. *IJSWIS* 10, 2 (2014), 63–86.

[19] Anna Sapienza, André Panisson, JTK Wu, Laetitia Gauvin, and Ciro Cattuto. 2015. Anomaly detection in temporal graph data: An iterative tensor decomposition and masking approach. In *AALTD*.

[20] Asara Senaratne, Peter Christen, Graham Williams, and Pouya G Omran. 2022. Unsupervised Identification of Abnormal Nodes and Edges in Graphs. *JDIQ* 15, 1 (2022), 1–37.

[21] Asara Senaratne, Pouya Ghiasnezhad Omran, Graham Williams, and Peter Christen. 2021. Unsupervised Anomaly Detection in Knowledge Graphs. In *IJCKG* (Thailand) *(IJCKG'21)*. ACM, New York, 161–165.

[22] Baoxu Shi and Tim Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-based systems* 104 (2016), 123–133.

[23] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *TKDE* 29, 12 (2017), 2724–2743.

[24] Yanhao Wang, Yuchen Li, Ju Fan, Chang Ye, and Mingke Chai. 2021. A survey of typical attributed graph queries. *World Wide Web* 24, 1 (2021), 297–346.

[25] Yaqing Wang, Fenglong Ma, and Jing Gao. 2020. Efficient Knowledge Graph Validation via Cross-Graph Representation Learning. In *CIKM* (Ireland) *(CIKM '20)*. ACM, New York, 1595–1604.

[26] Dominik Wienand and Heiko Paulheim. 2014. Detecting incorrect numerical data in dbpedia. In *ESWC*. Springer, Cham, 504–518.

[27] Yu Zhao, Huali Feng, and Patrick Gallinari. 2019. Embedding learning with triple trustiness on noisy knowledge graph. *Entropy* 21, 11 (2019), 1083.