Unsupervised Anomaly Detection in Knowledge Graphs

Asara Senaratne, Pouya Ghiasnezhad Omran, Graham Williams, Peter Christen
asara.senaratne@anu.edu.au
The Australian National University
Canberra, Australia

ABSTRACT

Anomalies such as redundant, inconsistent, contradictory, and deficient values in a knowledge graph are unavoidable, as such graphs are often curated manually, or extracted using machine learning and natural language processing techniques. Therefore, anomaly detection in knowledge graphs is an essential task that contributes towards its quality. Although there are approaches to detect anomalies in knowledge graphs, they are either domain dependent, not scalable to large graphs, or they require substantial human intervention. In this preliminary research paper we propose a novel unsupervised feature-based approach to anomaly detection in knowledge graphs. We first characterize triples in a directed edge-labelled knowledge graph using a set of binary features, and then use a one-class Support Vector Machine (SVM) to classify these triples as normal or abnormal. After selecting the features that have the highest consistency with the SVM outcomes, we provide a visualization of the identified anomalies, and the list of anomalous triples, thus supporting non-technical domain experts to understand the anomalies present in a knowledge graph. We evaluate our approach on the four knowledge graphs YAGO-1, KBpedia, Wikidata, and DSKG. This evaluation demonstrates that our approach is well suited to identify anomalies in knowledge graphs in an unsupervised manner, independent from the domain of the knowledge graph being evaluated.

CCS CONCEPTS

• **Computing methodologies → Anomaly detection**; **Unsupervised learning**; **Support vector machines**; • **Human-centered computing → Information visualization**; • **Information systems → Clustering**; **Graph-based database models**.

KEYWORDS

Data quality assessment; binary feature library; edge-labelled graphs; one-class classifier; visualization.

ACM Reference Format:
Asara Senaratne, Pouya Ghiasnezhad Omran, Graham Williams, Peter Christen. 2021. Unsupervised Anomaly Detection in Knowledge Graphs. In *The 10th International Joint Conference on Knowledge Graphs (IJCKG'21), December 6–8, 2021, Virtual Event, Thailand*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3502223.3502246

IJCKG'21, December 6–8, 2021, Virtual Event, Thailand

ACM ISBN 978-1-4503-9565-6/21/12…$15.00
https://doi.org/10.1145/3502223.3502246

1 INTRODUCTION

When constructing a Knowledge Graph (KG), it can either be manually curated like WordNet [7], manually generated by volunteers like Wikidata [23], automatically extracted from semi-structured text via hand-crafted or learned rules like YAGO [21], or automatically extracted from unstructured text using a machine learning technique like NELL [16]. Irrespective of the approach followed, the presence of anomalies is inevitable [3, 14, 15] when it is a human that adds content to a KG, or to any other source that is used to extract data, given that no human is omniscient. Similarly, machine learning techniques unlikely always produce completely accurate outcomes. It is therefore impractical to have a fully accurate and complete structured repository of knowledge. There will always be a trade-off between coverage and correctness, which is achieved differently in each KG [17].

To address these drawbacks, various methods for KG refinement have been proposed with the aim of improving KG quality. Similar to data quality dimensions [18], there exists a set of dimensions to assess the quality of KGs [8]. In the KG construction life cycle, there are two approaches named KG validation and KG enrichment that correspond to the two quality dimensions accuracy and completeness of KGs respectively. With the introduction of anomaly detection in KGs, we can achieve another two dimensions named concise representation and consistent representation [8].

Anomaly detection is important due to its ability to improve the quality of a KG. Performing anomaly detection as the first step will provide a much cleansed KG as an input to validation and enrichment methods, thereby improving their performance. KGs act as the knowledge repository for many applications such as personal assistants, software agents, search engines, question-answering machines, and many more [1]. Therefore, to ensure a quality output from these applications, it is important that their underlying KG is free from errors and anomalies, complete, and also has a concise and consistent representation of knowledge.

Contributions: In this preliminary research paper we introduce an unsupervised anomaly detection approach for KGs. Our approach develops a feature library which has a set of pre-defined binary features that we can apply on any KG irrespective of its domain and size. The features consider aspects such as structural arrangements of a graph, frequency of occurrence of subjects, predicates and objects, entity types of the subject and object, data type conformity of literals, duplicate triple occurrences, applicability of a predicate with a given entity type, and so on. These features are capable of identifying anomalies pertaining to data quality of the triples while also identifying any inconsistencies or contradictions among the triples.

Using this feature library, we construct one binary feature vector for each triple on the KG thus forming a matrix of features. We use a one-class SVM to learn the feature matrix where we then identify

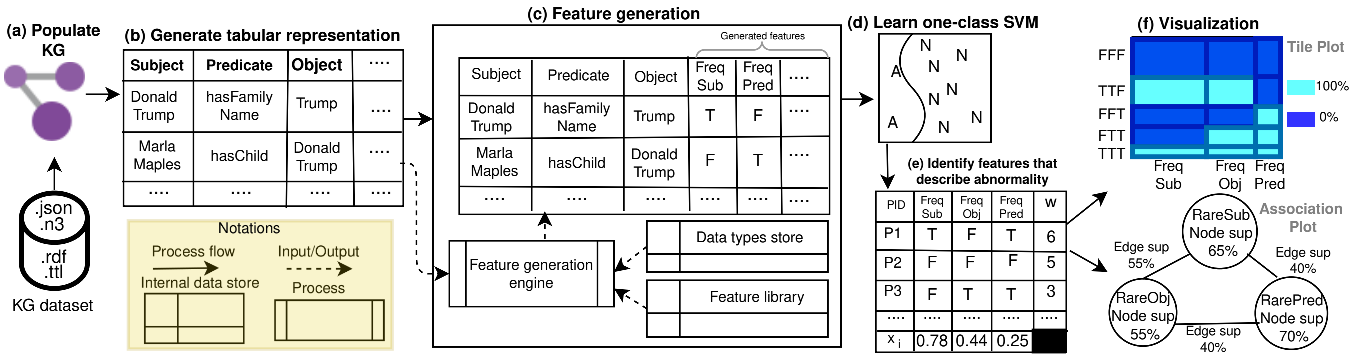

Figure 1: Overview of the anomaly detection process to identify anomalous triples in a KG, as described in Section 3.

the highly consistent features as the prominent set of features in determining the anomalous aspects of the KG. We present the anomalous patterns found in triples graphically to provide a visual representation of the anomalies existing in the KG. We evaluate our approach on four open KGs.

We can describe the novelty of our approach using its capabilities. Our approach is capable of detecting a multitude of anomalies such as duplicate triples, data type errors, contradicting triples, and entities with missing or incomplete data. The triple pair <Marla_Maples, hasChild, Donald_Trump> and <Marla_Maples, isMarriedTo, Donald_Trump> is an example of a contradicting triple in YAGO-1 that was identified by our approach. Also, our approach is unsupervised which means no human engagement is required to detect anomalies. Furthermore, our approach is domain and format independent. The use of a feature library that is independent from the KG makes our approach applicable on any KG irrespective of its domain and storage type.

2 RELATED WORK

Attributed graphs are rich in information, and therefore recently the paradigm of anomaly detection in graphs has shifted to attributed graphs [5]. CODA [9] is a generative model that detects outliers within the context of a community such that the identified outliers deviate significantly from the rest of the members in the community. MIDAS [2] is aimed at detecting micro cluster anomalies given a stream of edges from a dynamic graph, where an anomaly score is assigned to edges in an online manner as opposed to individual edge detection. GraphUCB [5] is an interactive anomaly detection algorithm defined on attributed networks. It presents a contextual multi-armed bandit algorithm that interactively incorporates human feedback on the identified anomalies.

The main difference between our proposed approach and the above discussed approaches is that we focus on edge-labelled graphs instead of attributed or dynamic graphs. Also, our approach does not interact with users during the anomaly detection process. While our definition of an anomaly is similar to CODA [9], we are interested in identifying anomalous triples considering the entire KG as opposed to communities or clusters.

In the context of anomaly detection in KGs, the pattern discovery and anomaly detection approach proposed by Jia et al. [13] presents a reasoning system to discover abnormal patterns and unusual activities given a stream of unstructured data such as tweets. Felfernig et al. [6] proposes an interface named ICONE to support knowledge engineers to develop and maintain Configuration Knowledge Bases (CKB) with capabilities to visually represent anomalies. ICONE is capable of detecting inconsistencies and redundancies in CKBs. Wienand et al. [25] introduces an unsupervised numerical outlier detection method to identify incorrect numerical values in DBpedia. Zhang et al. [26] studies the problem of discovering exceptional facts about entities in KGs. An entity is exceptional among the entities in the context under consideration, such as movies, music, or people. Jabeen et al. [12] presents an approach that can perform adaptive outlier detection against the cohorts of classes the data represent, where a cohort is a set of classes that are similar based on a set of selected properties.

Our approach differs from the above in the following ways: (1) Our approach is KG and data type independent unlike [12, 25]. (2) Unlike [6, 26], our work can detect a multitude of anomalies. (3) Even though our approach focuses on structured data, by changing only the features of the feature library, we can apply our approach on unstructured data to detect anomalies similar to [13].

3 PROPOSED METHODOLOGY

Following recent work in anomaly detection in KGs [6, 13, 25], our aim is to discover abnormal triples, on the basis that they are rare, missing, inconsistent, duplicate, or incomplete. For example, a triple with a missing object value is considered as an abnormal triple. In this section, we describe in detail the steps of our approach as outlined in Figure 1.

3.1 Populate knowledge graph

We consider a structured data source such as XML, JSON, or TTL, and convert it into a directed edge-labelled KG, $G = (V, E)$ containing a set of nodes (or vertices) V and a set of labelled edges E connecting these vertices.

The Resource Description Framework (RDF) is a standardised data model based on directed edge-labelled graphs with the W3C recommendation[1]. The RDF model defines three types of nodes in a graph such as Internationalized Resource Identifiers (IRIs) I which

[1]https://www.w3.org/TR/rdf-concepts/#section-Graph-URIref

assigns a global identifier for entities I_e and relations I_r on the web (where $I = I_e \cup I_r$), literals L which represents strings and other datatype values, and blank nodes B which are anonymous nodes (not a URI reference or a literal) that do not have an identifier [11]. We therefore have the node set $V = (I_e \cup L \cup B)$, and edge set $E \in V \times I_r \times V$.

Each edge $e \in E \in G$ is considered as a RDF triple. A triple (also named as a triplet) contains the three elements subject $s \in S$, predicate $p \in P$, and object $o \in O$. A triple is denoted as (s, p, o) where $(s, o) \in V$, and $(s \times p \times o) \in E$. Furthermore, $s \in (I, B)$, $p \in I_r$, and $o \in (I, L, B)$.

3.2 Generate tabular representation

In this step of our approach, the directed edge-labelled KG G is converted into a $n \times m$ matrix $\mathbf{T}$, where $n = |E|$ and $m = 3$, and the edge set $E \in G$ is represented in the form (s, p, o). The created $\mathbf{T}$ is further split into two matrices as $\mathbf{T}_l$ and $\mathbf{T}_e$ based on the type of the object node, such that $\mathbf{T} = \mathbf{T}_l \cup \mathbf{T}_e$. If $o \in L$, then $(s, p, o) \in \mathbf{T}_l$, else if $o \in I_e$, then $(s, p, o) \in \mathbf{T}_e$. This separation is made because we generate different sets of features for triples with a literal as the object, and for triples with an IRI object.

3.3 Feature generation

Our approach introduces a feature library with a set of pre-defined binary features. Presence of duplicate triples, object's data type mismatches the data type implied by the predicate, frequent subject/object in/out degree, frequent entity type of subject/object are some of the examples of features in the feature library.

Based on the set of feature generation functions $\mathcal{F}$ in the feature library, $\mathcal{F}_l$ for $\mathbf{T}_l$ and $\mathcal{F}_e$ for $\mathbf{T}_e$ where $\mathcal{F} = \mathcal{F}_l \cup \mathcal{F}_e$, we generate two feature matrices $\mathbf{F}_l$ and $\mathbf{F}_e$, such that $\mathcal{F}_l$ determines the feature matrix $\mathbf{F}_l$, and $\mathcal{F}_e$ determines the feature matrix $\mathbf{F}_e$. While the feature library stores all feature generation functions $\mathcal{F}$, the feature generator is responsible for: (1) retrieving the matrix of triples $\mathbf{T}_e$ or $\mathbf{T}_l$, data types store D, the feature generation functions $\mathcal{F}$ from the feature library as inputs, and (2) generating the output $\mathbf{F}_e$ or $\mathbf{F}_l$ as depicted in Step (**c**) of our approach in Figure 1.

In $\mathbf{F}_l$, we have one feature vector $\mathbf{f}$ per triple in $\mathbf{T}_l$ (the number of rows of $\mathbf{F}_l$ is $|\mathbf{T}_l|$ and the number of columns is $|\mathcal{F}_l|$). Similarly, in $\mathbf{F}_e$, we have one feature vector $\mathbf{f}$ per triple in $\mathbf{T}_e$ (the number of rows of $\mathbf{F}_e$ is $|\mathbf{T}_e|$ and the number of columns is $|\mathcal{F}_e|$).

3.4 Learn one-class SVM

We next train several one-class Support Vector Machine (ν-SVM) classifiers [20] to characterize feature vectors as normal or abnormal [4], where for each feature vector $\mathbf{f}$, a trained ν-SVM will provide a calculated anomaly score. To increase the robustness of our approach, we train several one-class ν-SVMs with different kernel functions [19]. We average the obtained abnormality scores to determine the b most abnormal feature vectors, where $b > 1$ is the *budget* defined by a decision maker indicating the maximum number of abnormal vectors he/she could investigate further. At the end of this step, the feature matrix ($\mathbf{F}_l$ or $\mathbf{F}_e$ under consideration) will contain b feature vectors labeled as abnormal while all others are labeled as normal.

Table 1: KG summaries.

KG	$\lvert I_e \rvert$	$\lvert \mathbf{T}_l \rvert$	$\lvert \mathbf{T}_e \rvert$
YAGO-1	2,215,094	21,337,521	922,741
KBpedia	62,796	534,032	227,060
Wikidata	14,036,475	53,541,372	51,559,889
DSKG	5,952	22,202	828,086

3.5 Identify features that describe abnormality

We next obtain, for each feature, a score of how consistent its values (of $True$ or $False$) in the feature vectors compared to the ν-SVM classification outcome (of normal and abnormal). For example, if a feature is always $True$ for those feature vectors labeled as abnormal and $False$ otherwise, then this feature will be highly useful in describing anomalous behavior. We select the top features with the highest such consistency scores, and using only these features we generate all binary patterns from the feature vectors labeled as abnormal. We do not consider the feature vectors labelled normal as we believe they do not require any corrections.

3.6 Visualization

As the final step, we visualize the anomalous binary patterns via a tile plot [22], and the associations among these highly consistent features via an association plot [24]. The tile plot shows how the selected features occur in different binary patterns of the anomalous triples. The size of a tile is determined by the consistency score of a feature (tile length), and how frequently a certain pattern occurs in the group of abnormal feature vectors (tile height). The color intensity of a tile represents the binary value of the respective feature in that particular pattern. To improve visualization, if too many binary patterns are generated, we apply agglomerative clustering [10] to group similar binary patterns. Then, each row of the tile plot represents a cluster of binary patterns.

An association plot, on the other hand, shows the features with the highest edge support between them, as calculated based on the maximum ratio of $True$ or $False$ values in the set of abnormal feature vectors. Combined, these two types of visualizations can help a user in understanding the features that contribute most to the anomalous characteristics of their KG.

4 EXPERIMENTAL EVALUATION

We implemented our approach using Python version 3. All experiments were run on a server with 64-bit Intel Xeon (2.4 GHz) CPUs, 512 GBytes of memory, and Ubuntu 18.04. The program code is available on Github (see: https://github.com/AsaraSenaratne/anomaly-detection-kg). We performed the experiments using the four KGs YAGO-1[2], KBpedia[3], Wikidata[4], and DSKG[5]. A summary of these four KGs are provided in Table 1.

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

Table 2: Examples of anomalous triples.

KG	Subject	Predicate	Object	Reason for Anomaly
KBpedia	2011IIHFWorld Championship	altLabel	2011 ice hockey world chamionship	Duplicate triples but with a spelling mistake in the object of the first triple.
	2011IIHFWorld Championship	altLabel	2011 ice hockey world championship	
	AgriculturalAircraft	altLabel	Crop-duster	A subject with three alternative labels conveying the same information.
	AgriculturalAircraft	altLabel	Cropduster pilots	
	AgriculturalAircraft	altLabel	Crop-duster pilots	
DSKG	distribution/100236	byteSize	""	Missing object.
	dataset/1816	describedIn	https://github.com/bgsu-rna/rnao	One dataset with three resources for descriptions is rare in the KG.
	dataset/1816	describedIn	http://soc.southalabama.edu/huang/papers/BIBM-15-1.pdf	
	dataset/1816	describedIn	http://omnisearch.soc.southalabama.edu/w/index.php/Ontology	
YAGO-1	Marla_Maples	hasChild	Donald_Trump	Contradicting triples.
	Marla_Maples	isMarriedTo	Donald_Trump	
	Raymond_Dalmau	bornOnDate	1950-##-##	Missing information in object.
Wikidata	L17778	altLabel	0	Abnormal object value. The subject 'L17778' referring to the English term "Lank" has the alternate label "0".
	L158675-F1	altLabel	""	Missing object value.

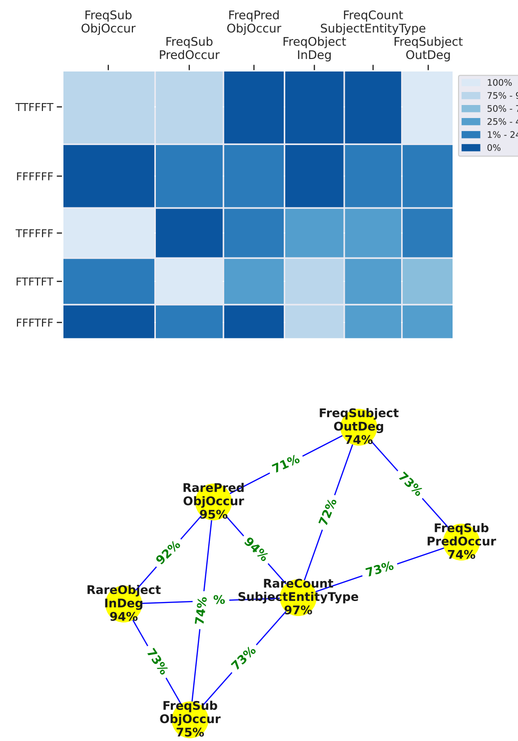

Figure 2: Tile and association plots of T_e of YAGO-1.

4.1 Results and discussion

We now present the visualizations obtained for one of the KGs (YAGO-1), and provide a few examples of the interesting anomalous triples obtained for all four KGs in Table 2. We manually verified the triples in this table to ensure their anomalous nature.

Figure 2 shows the identified abnormal patterns and associations of the YAGO-1 triples in $\mathbf{T}_e$. The triples of *Marla Maples* and *Donald Trump* in Table 2 are abnormal because in the real world, two people cannot hold the relationships mother-son and husband-wife simultaneously. These two triples are represented by the third cluster (row) of the tile plot in Figure 2. The triples belonging to the third cluster have a subject and object pair occurring together frequently (first tile), hence a tile with a lighter intensity. The fact that *Marla Maples* and *Donald Trump* having more than one relationship among them is rare to see among other triples of people.

The second tile of the third cluster provides another anomalous feature of the triples belonging to that cluster. With a darker color intensity, it implies that the occurrence of a particular subject and predicate together is rare. The subject and predicate combinations <*Marla Maples*, *hasChild*>, and <*Marla Maples*, *isMarriedTo*> have rare occurrences as the KG does not contain much information about *Marla Maples*. This is further confirmed by the sixth tile stating rare subject-out-degree. Also, the predicate and object occurrence together is rare because *Donald Trump* mostly appears as the subject and rarely as the object in triples. Hence, a rare object-in-degree as well. In YAGO-1, there are more triples about movies and song albums compared to triples about people. Hence, there is a rare count of occurrence of subject's entity type for triples about people. As *Marla Maples* belongs to the entity type *PERSON*, the two example triples are considered rare within the KG.

As per the association plot, 97% of the triples are anomalous due to the rare occurrence of subject's entity type (for example, YAGO-1 has less information about places compared to movies and song albums), 74% are abnormal due to subject and predicate of a triple frequently occurring together (as in the case of duplicate triples), and 73% are abnormal due to both these features.

5 CONCLUSION AND FUTURE WORK

In this preliminary research paper we study the problem of discovering anomalous triples in a knowledge graph (KG) in an unsupervised manner. The approach uses a feature library that contains features pertaining to structure of a graph, frequency of occurrence of subjects, predicates and objects, entity types of the subjects and objects, data type conformity of literals, duplicate triple occurrences, applicability of a predicate with a given entity type, and so on. A triple is considered anomalous if it has missing, duplicate, contradicting, incomplete, or inconsistent data. Unlike other work, our approach is domain independent, has no input type limitations, has an ability to identify a multitude of anomalies without depending on external sources, and can compliment KG validation and enrichment. We intend to improve our work using two approaches. First, we will perform manual evaluation. Second, we will introduce noise to KGs to determine how well our approach detects them. As for future work, we aim to make automatic suggestions on possible means of correcting the anomalies identified.

REFERENCES

[1] Caleb Belth, Xinyi Zheng, Jilles Vreeken, and Danai Koutra. 2020. What is normal, what is strange, and what is missing in a knowledge graph: Unified characterization via inductive summarization. In *Proceedings of The Web Conference 2020*. ACM, New York, USA, 1115–1126.

[2] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. 2020. Midas: Microcluster-based Detector of Anomalies in Edge Streams. In *Conference on Artificial Intelligence*, Vol. 34. AAAI Press, California, USA, 3242–3249.

[3] Antoine Bordes and Evgeniy Gabrilovich. 2014. Constructing and mining web-scale knowledge graphs: KDD 2014 tutorial. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, USA, 1967–1967.

[4] Peter Christen, Ross W. Gayler, Khoi-Nguyen Tran, Jeffrey Fisher, and Dinusha Vatsalan. 2016. Automatic Discovery of Abnormal Values in Large Textual Databases. *Journal of Data and Information Quality* 7, 1–2 (2016), 1–35.

[5] Kaize Ding, Jundong Li, and Huan Liu. 2019. Interactive Anomaly Detection on Attributed Networks. In *International Conference on Web Search and Data Mining*. ACM, New York, USA, 357–365.

[6] Alexander Felfernig, Florian Reinfrank, and Gerald Ninaus. 2012. Resolving anomalies in configuration knowledge bases. In *International Symposium on Methodologies for Intelligent Systems*. Springer, Berlin, Heidelberg, 311–320.

[7] Christiane Fellbaum. 2010. WordNet. In *Theory and applications of ontology: computer applications*. Springer, Dordrecht, Netherlands, 231–243.

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

[9] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. 2010. On Community Outliers and Their Efficient Detection in Information Networks. In *International Conference on Knowledge Discovery and Data Mining*. ACM, New York, USA, 813–822.

[10] Jiawei Han and Micheline Kamber. 2006. *Data Mining: Concepts and Techniques* (2 ed.). Morgan Kaufmann, Chicago.

[11] 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. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–37.

[12] Hajira Jabeen, Rajjat Dadwal, Gezim Sejdiu, and Jens Lehmann. 2018. Divided we stand out! Forging Cohorts fOr Numeric Outlier Detection in large scale knowledge graphs (CONOD). In *European Knowledge Acquisition Workshop*. Springer, Switzerland, 534–548.

[13] Bin Jia, Cailing Dong, Zhijiang Chen, Kuo-Chu Chang, Nichole Sullivan, and Genshe Chen. 2018. Pattern Discovery and Anomaly Detection via Knowledge Graph. In *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, New Jersey, United States, 2392–2399.

[14] Shengbin Jia, Yang Xiang, Xiaojun Chen, and Kun Wang. 2019. Triple trustworthiness measurement for knowledge graph. In *The World Wide Web Conference*. ACM, New York, USA, 2865–2871.

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

[16] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Commun. ACM* 61, 5 (2018), 103–115.

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

[18] Leo L Pipino, Yang W Lee, and Richard Y Wang. 2002. Data quality assessment. *Commun. ACM* 45, 4 (2002), 211–218.

[19] Ibai Roman, Roberto Santana, Alexander Mendiburu, and Jose A Lozano. 2020. In-depth Analysis of SVM Kernel Learning and its Components. *Neural Computing and Applications* 33 (2020), 1–20. Issue 10.

[20] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation* 13, 7 (2001), 1443–1471.

[21] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *European Semantic Web Conference*. Springer, Cham, Switzerland, 583–596.

[22] Viswa Viswanathan, Shanthi Viswanathan, Atmajitsinh Gohil, and Chiu David Chiu Yu-Wei. 2016. *R: Recipes for Analysis, Visualization and Machine Learning*. Packt Publishing Ltd, Birmingham, UK.

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

[24] Xin Wang, Yang Xu, and Huayi Zhan. 2020. Extending Association Rules with Graph Patterns. *Expert Systems with Applications* 141 (2020), 112897.

[25] Dominik Wienand and Heiko Paulheim. 2014. Detecting incorrect numerical data in dbpedia. In *European Semantic Web Conference*. Springer, Cham, Switzerland, 504–518.

[26] Gensheng Zhang, Damian Jimenez, and Chengkai Li. 2018. Maverick: Discovering exceptional facts from knowledge graphs. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, New York, USA, 1317–1332.