Secondary Publication

Schramm, Simon; Wehner, Christoph; Schmid, Ute

Comprehensible Artificial Intelligence on Knowledge Graphs : A survey

Date of secondary publication: 08.09.2025

Submitted Version (Preprint), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-110308x

Primary publication

Schramm, Simon; Wehner, Christoph; Schmid, Ute (2024): Comprehensible Artificial Intelligence on Knowledge Graphs : A survey, in: arXiv, S. 1–23, doi: 10.48550/arxiv.2404.03499.

Legal Notice

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available under a Creative Commons license.

The license information is available online:
https://creativecommons.org/licenses/by/4.0/legalcode

arXiv:2404.03499v1 [cs.AI] 4 Apr 2024

Comprehensible Artificial Intelligence on Knowledge Graphs: A Survey.

Simon Schramm[a,b,1], Christoph Wehner[a,1], Ute Schmid[a]

[a]*Cognitive Systems Group, University of Bamberg, An der Weberei 5, Bamberg, 96049, Bavaria, Germany*
[b]*BMW Group, Petuelring 130, Munich, 80809, Bavaria, Germany*

Abstract

Artificial Intelligence applications gradually move outside the safe walls of research labs and invade our daily lives. This is also true for Machine Learning methods on Knowledge Graphs, which has led to a steady increase in their application since the beginning of the 21*st* century. However, in many applications, users require an explanation of the AI's decision. This led to increased demand for Comprehensible Artificial Intelligence. Knowledge Graphs epitomize fertile soil for Comprehensible Artificial Intelligence, due to their ability to display connected data, i.e. knowledge, in a human- as well as machine-readable way. This survey gives a short history of Comprehensible Artificial Intelligence on Knowledge Graphs. Furthermore, we contribute by arguing that the concept of Explainable Artificial Intelligence is overloaded and overlaps with Interpretable Machine Learning. By introducing the parent concept Comprehensible Artificial Intelligence, we provide a clear-cut distinction of both concepts while accounting for their similarities. Thus, we provide in this survey a case for Comprehensible Artificial Intelligence on Knowledge Graphs consisting of Interpretable Machine Learning and Explainable Artificial Intelligence on Knowledge Graphs. This leads to the introduction of a novel taxonomy for Comprehensible Artificial Intelligence on Knowledge Graphs. In addition, a comprehensive overview of the research in this research field is presented and put into the context of the taxonomy. Finally, research gaps in the field are identified for future research.

Keywords: Knowledge Graphs, Comprehensible Artificial Intelligence, Explainable Artificial Intelligence, Interpretable Machine Learning

1. Overview

Artificial Intelligence (AI) applications gradually move outside the safe walls of research labs and invade our daily lives. Many real-world AI applications, e.g., in medicine [1], industrial product development [2, 3], and autonomous driving [4], are safety-critical. Thus, AI algorithms used in those fields are regulated [5]. The regulations aim to give the user the ability to comprehend the AIs decision [5]. However, most state-of-the-art AI methods, like deep neural networks, are too complex and complicated to comprehend for a user, so-called black box models. Leading to increased demand for Comprehensible Artificial Intelligence (CAI) research in the last years [6, 7].

CAI is a set of methods that enable stakeholders to understand and retrace the output of AI models [8, 9, 7]. CAI has two major sides. One side reconstructs the decision-making process of a black box model with the help of a human-understandable mapping from the model's input to its output. This side is called Explainable Artificial Intelligence (XAI) [8]. The other side of CAI aims to create AI models with, by default, human-understandable decision-making processes, so-called white boxes. This side is called Interpretable Machine Learning (IML) [8]. Section 2 provides an in-depth discussion of CAI.

On the necessity of Comprehensible Artificial Intelligence on Knowledge Graphs. Regularly, AI models with human-understandable decision-making processes are regarded as Ḣowever, the concept XAI is overloaded. For example, take *TEM* [10] and *RelEx* [11]. Both methods talk about explainability. While both satisfy similar high-level needs, e.g., justifying the model behavior, identifying risks, debugging the model, and uncovering unknown rules, they are fundamentally different.
One method (*TEM*) holds a complete Machine Learning (ML) model, with inherent interpretability by human-

[1]The authors contributed equally.

understandable inner mechanics.
The other method (*RelEx*) is a post-hoc explainability method applied to a black box model to discover which part of the input contributed to the output.
We aim to account for their differences by taxonomizing such methods in IML and XAI. And at the same time, we acknowledge their similarities in taxonomizing IML and XAI as the two components of CAI (cf. Figure 1).

Knowledge Graphs (KG) are human- and machine-readable representations of semantically linked data, referred to as knowledge over a particular domain [12]. Semantically linked data is predestined to create AI models with human-understandable decision-making processes [13, 7], which shall be the notion of CAI in this survey. This survey aims to give an overview of CAI methods on KGs.

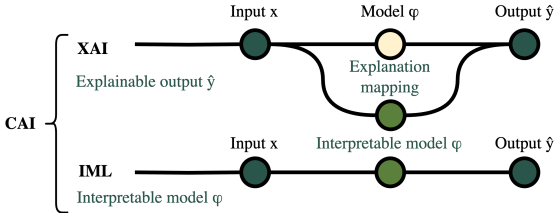


Figure 1: Generic CAI framework.

We start with an introduction to the history of CAI on KGs. Thereby, we define the term CAI and put the notions IML and XAI into context. Furthermore, we taxonomize state-of-the-art approaches to CAI on KGs by representation, task, foundation, and comprehensibility. Finally, opportunities for future research of CAI on KGs are laid out. The following section provides a brief history of CAI on KGs, followed by the proceeding and sources of the survey, the scope of the survey, and the related work.

1.1. A brief history of Comprehensible Artificial Intelligence on Knowledge Graphs

The first KGs were created as semantic networks for a university project from 1972 to 1980 [14]. Between 1985 and 2007 the idea of a KG was applied to different fields, such as human language in *Wordnet* [15]. Algorithms mainly focused on utilizing the symbolic semantics of the KG to reason and learn new rules over the domain of a KG [16, 17]. These algorithms, like *FOIL* [18], were fully comprehensible. The success of KGs rise abruptly and rapidly when *DBpedia* [19] and *Freebase* [20] were created as general-purpose KGs, which led to the *Google* KG in 2012 [21]. Since then, a plethora of private companies and academic institutions utilize KGs for a variety of applications [12].

Researchers introduced a wide range of ML methods operating on KGs, fueled by the ever-growing interest in KGs. This gave rise in the last decade to incomprehensible methods, like translational models [22] and Graph Neural Networks (GNN) [23]. However, the incomprehensibility of those algorithms prohibits their applications in high-stake applications. Thus, recently CAI [7] KGs has gained significant attention, which is also the focus of this survey.

1.2. Proceeding and sources

Following the systematic literature search process of Webster and Watson [24] and vom Brocke et al. [25], we identified 120 researchers from the community publishing papers in the domain of CAI and KGs.

We contacted our fellow researchers posing the following questions:
(1) Which papers do you think are crucial to cite in such a survey?
(2) Which major outlets (conferences/journals) have you seen published literature on Comprehensible Artificial Intelligence on Knowledge Graphs?
(3) Are there topics you would like to see discussed in such a survey?
(4) Where do you see future research directions for Comprehensible Artificial Intelligence on Knowledge Graphs?
The answers provided to these questions can be found in Appendix A. As a result of this questionnaire, we determined the search string *"Knowledge Graph*"* ***AND*** *"Machine Learning*"* ***OR*** *"Artificial Intelligence*"* and *"Knowledge Graph*"* ***AND*** *"Interpretable*"* ***OR*** *"Explainable*"* ***OR*** *"Comprehensible*"*, which we searched for in titles, abstracts, and keywords. Figure 2 displays the resulting 163 publications. Figure 2 reveals, that especially journal articles accumulate in the very recent past.

We utilized the *Scimago Institutions Rankings*[2], to assure a high quality in selecting publications. We focused on journals and conference proceedings above rank 150. However, some articles we cite are not published yet and are thus only available as preprint. Table 1 presents the most frequent journals, conferences or publishers of the literature list of this survey. Eventually, we chose 55

[2] *Scimago Institutions Rankings* is a ranking of scientific journals and conferences, based on a composite indicator that represents research performance, innovation outputs and societal impact measured by the web visibility of the respective journal or conference [26, 27, 28].

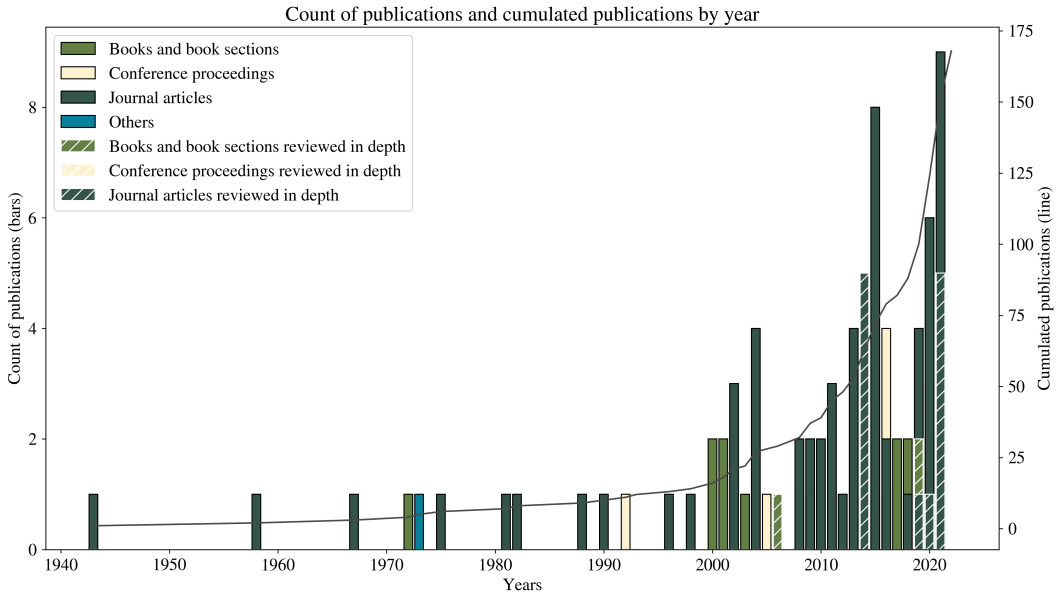


Figure 2: Reviewed publications by year.

publications from a total of 163 titles obtained by our structured search, based on an assessment of the publications' abstracts. In the following sections, the 55 publications chosen for close assessment will be described, and taxonmized according to Table 2. Later, a combination of the taxonomy provided by this survey with the discrimination of IML and XAI on KGs will be provided in Figure 10.

Type Publisher	Book (section)	Conference proceeding	Journal article
Springer	2	8	2
IEEE	0	0	1
Elsevier	0	0	2
ACM	0	13	0
Others	0	10	17

Table 1: Most frequent publishers by publication type.

The following Section 1.3 will outline the scope and the taxonomy of this survey.

1.3. Scope of the survey

Based on the typology of literature reviews developed by Paré et al. [29], the survey at hand aims at a narrative method to describe the state of the art of CAI on KGs. The motivation for this scope arises from the surging interest in KGs as input for ML methods and the mounting importance of comprehensibility in AI. Thereby, the survey follows the notion of CAI as a conglomerate of methods that reconstruct the decision-making process of a black box AI model with the help of a human-understandable mapping from the model's input to its output **and** methods that make the decision-making process of an AI model human-understandable [30, 31, 32, 33]. The concept CAI is necessary, as there has yet to be a clear-cut distinction between IML and XAI in past publications. Methods for human-understandable AI are generally referred to as XAI by previous publications. However, this leaves ML methods with inherent interpretability and post-hoc explanation methods in the same class. This survey aims to clarify the distinction between the concepts Interpretable Machine Learning and Explainable Artificial Intelligence, as well as to define the unifying term CAI. Since this scope entails both systems that learn explanations and systems that do not involve a learning process, it makes sense to refer to comprehensibility in the context AI, rather than ML. Hence, in Section 2.1, we will come to such a formal distinction, while summarizing both concepts by the notion of CAI.

In the context of CAI, a KG U can be used as input for ML, whereby data x_i forms the KG U that can

be embedded for being used in a ML model β. Other than that, KGs can also be defined as target output representations of an algorithm.[3] Figure 3 visualizes this simplified, generic KG-ML pipeline.

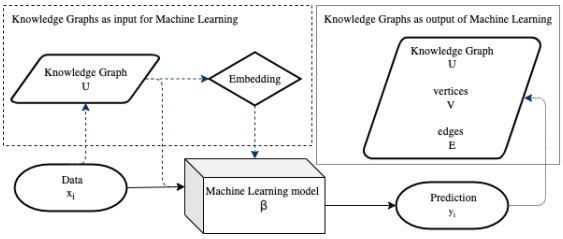


Figure 3: Simplified and generic KG-ML pipeline with KGs as input or output.

KGs as target output representations of a model β cannot render the model interpretable, nor can they epitomize an output that refers to the input data x_i in order to generate explainability. Henceforth, the focus of this survey is CAI on KGs as an input for ML.

In accordance with Webster and Watson [24], we developed a taxonomy for the publications reviewed, based on the kind of representation, the setting of the KG, the kind of optimization, the task of the AI model and the type of comprehensibility that is underlying. The taxonomy is shown in Table 2. A profound description of the scopes of our taxonomy will be provided in Section 2.

1.4. Related work

Several surveys exist related to CAI and KGs, although they usually refer to XAI instead of CAI.

For example, Tiddi and Schlobach [35] elaborate on CAI on KGs. They follow a broader understanding of CAI. Tiddi and Schlobach [35] consider methods as CAI that rationalize the decision of an AI model through external reasoning, independent of the AI model's decision-making process. In contrast, the survey at hand understands CAI solely as a method of transparent decision-making AI models.

Bianchi et al. [36] apply a similar understanding of CAI to the survey at hand. However, Bianchi et al. [36]'s survey focuses on a general overview of AI with KGs as an input, including comprehensible methods. In contrast, the survey at hand fully commits to CAI on KGs.

In addition, Lecue [37] elaborate on CAI on KGs. They structure their paper by the AAAI paper submission taxonomy, e.g., "Computer Vision" and "Game Theory". Each part of the taxonomy is looked at from the perspective of "Research Question", "XAI Challenge", "Methods", "Limitations", and "Opportunity". Their taxonomy gives researchers an excellent introduction to how CAI on KGs can be leveraged in different fields of AI, what the pitfalls are, and where future research directions lie. In contrast, the survey at hand aims to provide a complete overview of existing CAI methods on KGs and to cluster them in the taxonomy proposed in Section 1.3. Three other related surveys were identified. However, they focus on comprehensible recommender systems [38], comprehensible Graph Neural Networks [39], and comprehensible semantic web technologies [40], including non-KG-related ML methods.

The following section shall provide relevant theoretical background for the concepts that form our taxonomy (cf. Table 2).

2. Theoretical Background and Taxonomy of Comprehensible Artificial Intelligence on Knowledge Graphs

CAI, XAI as well as IML are sometimes used interchangeably [41, 11, 10]. We trace this back to the fact that especially the concept XAI is overloaded and, at the same time, not sharply differentiated from CAI and IML. Therefore, this section shall provide a structured, theoretical background on these notions and outline the concepts of our taxonomy in detail.

In previous literature, XAI describes inherently interpretable AI models like *TEM* [10] (cf. Table 4) and post-hoc explanation methods like *RelEx* [11] (cf. Table 5). This includes post-hoc explanation methods that give third-party rationalization of the AI model's decision [42, 35]. Meanwhile, there is the concept IML. IML models are inherently interpretable [8], which makes the concept overlap with a particular usage of XAI. The fuzzy border and the overload are due to similar high-level needs satisfied by IML and XAI, like identifying risks of the AI model's output or providing an ethical justification of the model's output [43]. However, inherently interpretable AI models and post-hoc explanations methods are fundamentally different. One is a complete ML model able to classify its input, and the other is a technique unable to classify on its own. Thus, both should be described with different concepts. This survey accounts for the fundamental differences in the inherently interpretable AI models and post-hoc explanation methods by grouping them in IML and XAI. A clear-cut definition for both concepts shall be provided

[3] Recent progress in computer vision serves as an vivid example, elaborating on techniques that correctly infer relations between detected objects, such as, *window on building*, or *building near person* [34].

Scope	Characterization
Representation	Symbolic (SB), Sub-symbolic (SSB), Neuro-symbolic (NSB).
Foundation	Factorization Machines (FM), Translational Learning (TL), Rule-based Learning (RBL), Neural Networks (ANN), Reinforcement Learning (RL), Others (O).
Task	Link Prediction (LP), Node Clustering (NC), Graph Clustering (GC), Clustering (C), Recommendation (R).
Comprehensibility	Interpretable Machine Learning (IML), Explainable Artificial Intelligence (XAI).

Table 2: Taxonomy of the survey.

in this section. In addition, this survey accounts for the similar high-level needs IML and XAI satisfy by introducing CAI as their parent concept in the papers' taxonomy.

This survey follows the understanding of the concepts XAI and IML as introduced in work by Rudin [8] and Schmid [9]. Composing the concept of XAI and IML to the overarching concept of CAI is also part of the contribution of this writing. The concepts CAI, XAI, and IML shall be outlined in the following sections.

2.1. Comprehensibility

CAI is about opening up black box models and explaining input, inside reasoning and processing, and output in an adequate manner to the relevant users. CAI makes a black box AI model transparent without decreasing its performance. Another side of CAI is developing algorithms that produce interpretable ML models, so-called white-box models. Those white-box models are supposed to perform close to the state-of-the-art, in order to reach widespread application [43].

CAI serves the following four reasons.

(1) The CAI model is supposed to justify its output [43, 44]. One possible application of this is to verify that the resulting output aligns with the ethical principles of a particular moral framework [43, 44].

(2) Furthermore, explanations help to identify risks and flaws of an AI model. A possible risk is the *Clever Hans* bias described by Lapuschkin et al. [45]. They describe an image classifier that classifies horses correctly because all pictures with horses have a particular watermark on them. The explanation of the classifier uncovers this flaw [43, 44].

(3) Finding risks and flaws of an AI model enable researchers or developers to debug and improve the model [43, 44].

(4) In addition, producing explanations for an AI model might uncover unknown rules discovered by the AI model [43, 44].

Figure 1 displays the generic CAI framework separating XAI from IML. The following shall elaborate more on the notions XAI and IML, contributing to a clear-cut notion of CAI.

2.1.1. Interpretability

IML methods require a per default human-understandable decision-making process to integrate the user into the system, rather than offering a set of descriptive tools [46, 30, 8, 9]. Thus, a IML model is able to classify and to provide a meaningful explanation for a classification by itself. The explanation is faithful to the model's decision-making process and behaviour. Such methods, shall, for the scope of this survey, be referred to as IML systems. XAI will be outlined in the following.

2.1.2. Explainability

While IML tries to uncover the whole decision-making process of the ML model, XAI is about mapping the input of a black box model to its output. That way, XAI methods compute an explanation of the ML model's behaviour. As the explanation method is external to the ML model, faithfulness is not guaranteed. However, the quality of a XAI method is measured by how faithful their explanations reflect the model's behaviour. The inside reasoning and processing of the black box model are not be made transparent by the XAI method.

A common theme in XAI is the attribution of a metric that represents the impact of a specific input characterization, frequently referred to as the attribution value of an input feature. The attribution tells how important an input feature is for or against a classification. Such methods are so-called attribution methods [47, 48, 49, 50]. Other innovative themes exist in XAI, like counterfactual explanation methods [51] or explanations methods based on Inductive Logic Programming (ILP) [52].

CAI significantly benefits from semantic information. KGs provide semantic information. Thus, the following sections shall introduce the KGs and the taxonomy of CAI on KGs.

2.2. Knowledge Graphs

A KG is a directed labeled graph [12] (cf. Figure 4).

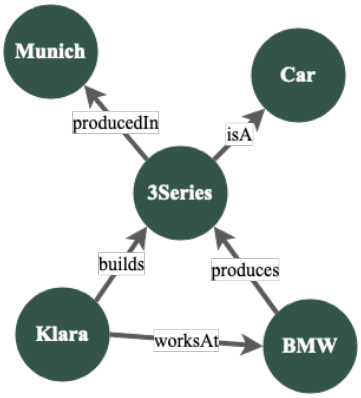


Figure 4: Car manufacturer KG.

In general, a graph is a set of connected entities. Entities are referred to as vertices v (or nodes), e.g., "*Klara*", while connectors are referred to as edges e (or relations), e.g., "*builds*", in the context of a graph. Thus, a graph U consists of a set of vertices $V = \{v_1, ..., v_i, ..., v_n\}$ and edges $E = \{e_1, ..., e_j, ..., e_m\}$. This renders a graph U as a subset of $V \times E \times V$ [53].

A directed graph is a graph in which edges have a direction. Thus, given a directed edge e_1 from vertex v_1 to vertex v_2, graph traversal via e_1 is exclusively possible from v_1 to v_2 and not vice versa [54]. A labeled graph is a graph in which holds $\forall v_i \in V$ and $\forall e_j \in E$ that exactly one element from a set of symbols $S = \{s_1, ..., s_k, ..., s_o\}$ is assigned to them. A symbol s_k is, in general, the name of an entity like "*Klara*" or the name of a relationship like "*worksAt*" [55]. A head entity $h \in V$ connected via a relation $r \in E$ to a tail entity $t \in V$, e.g., "*Klara worksAt BMW*", is called a triple. Due to the latter properties, a KG is optimal for storing highly relational data while preserving its semantics.

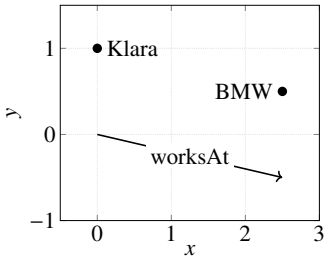
2.3. A taxonomy of Comprehensible Artificial Intelligence on Knowledge Graphs

The section introduces the critical aspects of CAI on KGs. First, the representations of KGs shall be discussed, followed by the tasks, and foundations of CAI on KGs.

2.3.1. Representations

KGs as inputs to AI models are represented sub-symbolically, symbolically, or neuro-symbolically [12].

The sub-symbolic representation of KGs gained vast popularity with the rise of scalable neural ML. The general idea is to embed the entities and relations of the KG into low dimensional numeric space, like an adjacency matrix or a real-valued matrix [22]. An example of entities and relations in $\mathbb{R}^2$ is depicted in Figure 5. However, at scale, such representations become incomprehensible for humans.



$$Klara = (0.0, 1.0)$$
$$BMW = (2.5, 0.5)$$
$$\vec{worksAt} = (2.5, -0.5)$$

Figure 5: Representation of the entities and relation, *Klara*, *BMW*, and *worksAt* in $\mathbb{R}^2$ $(x, y \in \mathbb{R})$.

A KG is by default a symbolic structure, as the nodes and edges of a KG are labelled with symbols. Thus, unlike the sub-symbolic representation, no specific mapping is required to achieve a symbolic representation of a KG. The symbolic representation is human comprehensible and allows for reasoning, inconsistency checking, and various other symbolic AI methods (cf. Section 3.1).

Combining sub-symbolic and symbolic representations, while leveraging the advantages of both representations, achieves a neuro-symbolic[4] representation [58] (cf. Section 3.1).

2.3.2. Tasks

We differentiate five different tasks of CAI on KGs, namely Link Prediction (LP), Node Clustering (NC), Graph Clustering (GC) and Recommendation (R).

Clustering can be divided into vertices-clustering [59] and structural clustering [60, 61]. Vertices-clustering clusters vertices by their similarity, like the bi-section k-modes clustering [62]. In contrast, structural clustering uses the graph topology as input to cluster the whole graph, like the *Louvain* clustering algorithm [63].

The recommendation task is about finding a set of relevant items for a user [64]. Users and items may

[4]Neuro-symbolic AI is also known as the third wave of AI [56, 57].

be modeled as entities of the KG. Liu and Duan [64] provide a comprehensive survey on KG-based recommender systems.

KGs are notoriously incomplete, i.e, relations between two entities might be missing. Predicting such missing relations, e.g., "*Klara worksIn Munich*", is called link prediction [22].

To classify a vertice in the KG regarding a set of classes external to the KG is called vertice classification [65], e.g., "*Klara*" is an "*Employee*".

Furthermore, it might be interesting to classify the KG as a whole. Molecules, for example, can be represented as KGs [66]. Graph classification is about classifying whether the molecule is toxic [67].

The following section shall give short examples of how foundations for those tasks are designed.

2.3.3. Foundations

There are numerous and diverse foundational ML methods for CAI on KGs. Only a few of them can be discussed due to the scope of the paper. Translational öearning, Neural Networks, in particular Graph Neural Networks, and rule-based learning are some of the most influential foundation (cf. Section 1.4), which are introduced in the following.

Translational learning methods translate the entities and relations of a KG into numeric space. This is achieved by learning a score function. The score function expresses a predefined property of the translational method [68]. For example, the score function of *TransE* is $d(h + r, t) = \|h + r - t\|$ [22]. The score function is optimized by a loss function to embed every element of a triple $(h, r, t) \in \mathbb{R}^2$, such that the relation added to a head entity results in the representation of the tail entity (cf. Figure 5)[22].

Popular connectionism methods are Neural Networks and Graph Neural Networks in the context of graphs. Graph Neural Networks are non-linear functions in the form of layered networks that can be optimized by gradient descent on all subsets of the graph U to perform regression or to learn classes or concepts [23]. However, standard Graph Neural Networks cannot integrate labels and directions of relations [23].

Rule-based learning methods heuristically search through possible combinations of logical clauses [69, 70, 71]. The resulting clauses, i.e. rules, optimize the coverage of triples, predictive accuracy, and other task-specific goal functions [72].

Other foundations exist, like Factorization Machines (FM) and Reinforcement Learning (RL). The authors refer to Bokde et al. [73] and Sutton and Barto [74] for a comprehensive introduction to those methods.

The components by which we taxonomize CAI on KGs are completely presented. In the following section, specific methods shall be discussed in accordance with our taxonomy.

3. Comprehensible Artificial Intelligence on Knowledge Graphs

We identified different lines of research in IML and XAI on KGs during the literature review (cf. Figure 6). There are three lines of research in IML on KGs, which are

1. Rule mining methods,
2. Pathfinding methods, and
3. Embedding methods,

and four lines of research in XAI on KGs, which are

1. Rule-based learning methods,
2. Decomposition-based methods,
3. Surrogate methods, and
4. Graph generation methods.

IML and XAI on KGs are fundamentally different but similar in the high-level needs they satisfy. The fundamental difference leads to unique lines of research in both concepts, and a varying number in lines of research this survey identified for both concepts.

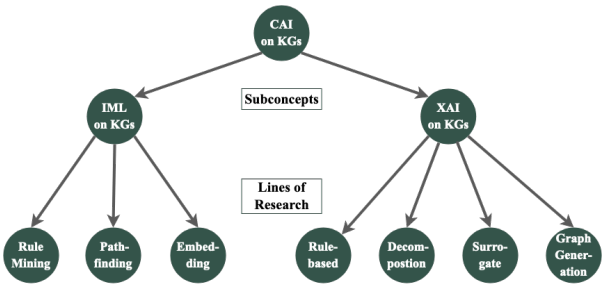


Figure 6: Hierarchy of the lines of research in CAI on KGs.

The following subsections shall introduce the lines of research in IML and XAI on KGs. The lines of research are described by presenting relevant. The aim is to draw a picture of how the research lines evolved and where past and current challenges lie.

3.1. Interpretable Machine Learning on Knowledge Graphs

IML methods on KGs come in many shapes and colors. As argued in 2.1.1, the main commonality of such models is that they are self-explanatory white-box models for prediction and classification tasks on KGs. This

survey identifies several lines of research in the field of IML on KGs. One of the earliest lines of research are rule mining methods.

3.1.1. Rule mining methods

One may trace back the roots of rule mining methods on KGs to ILP [16] algorithms like *FOIL* [18]. However, their scope is learning rules from any logical knowledge base. Thus, traditional ILP algorithms are not optimized to scale to large KGs [69, 70, 71]. Rule mining methods customized to IML on KGs improve the scalability.

One of the first and most impactful white-box rule mining algorithms specifically designed for KGs was *PRA* [71]. *PRA* samples numerous path types. Next, random walks through the KG are conducted. The random walks are constrained to follow one of the previously sampled path types. The random walks count the frequency of vertex-pairs being the start and end of the same path type. Finally, the frequencies of possible path types over the vertices are matched via logistic regression. That way, general rules, i.e. interpretable clauses (cf. Table 3), are inferred. The clauses may be used to predict missing links between vertices. Furthermore, the path type frequencies might be used as the embedding of a vertex [71]. While *PRA* has increased scalability, random walks are still computationally expensive. In addition, the link prediction accuracy is not state-of-the-art since factorization and translation methods, like *RESCAL* [75] and *TransE* [22], were introduced. Research following *PRA*, like *SWARM* [76], *RLvLR* [77], *RDF2rules* [78], and *AIME+* [72] on rule-based algorithms addresses those issues. In particular, *AIME+* is another ILP-inspired algorithm for scalable rule mining and link prediction on KGs. *AIME+* builds rules for link prediction by iteratively adding new clauses to a rule such that the approximated coverage and confidence of the new rule is maximized [72]. *AIME+* improved on scalability and predictive performance compared to *PRA*. Ontology Pathfinding (*OP*) [79], and *ScaLeKB* [80], scales the idea of *AIME+* to large benchmark KGs datasets like *Freebase*. It does so by parallelizing join queries, breaking the rule mining task into smaller sub-tasks, and eliminating unsound and resource-consuming rules before extending them. In addition, *AnyBURL* [81, 82] and subsequently *SAFRAN* [83] build on *PRA* to increase its scalability by an anytime bottom-up technique for rule learning [81] and aggregating rules through a scalable clustering algorithm [83]. These techniques allow rule mining methods, like *AnyBURL* and *SAFRAN*, to scale to modern large-scale KGs while showing state-of-the-art predictive performance [84].

One significant research effort was to scale symbolic rule mining approaches to large KGs, which was subsequently achieved by highly optimized heuristics and rule learning techniques. The following methods take a different path to scale rule mining to large KGs. They combine symbolic and sub-symbolic representations of the KG to efficiently mine interpretable rules. *NeuralLP* [85] mines interpretable rules from KGs. *NeuralLP* is inspired by the probabilistic deductive database *TensorLog* [86]. *NeuralLP* learns weighted chain-like logical rules. Those link the head of a query to its tail. It does so by embedding relations in an end-to-end differentiable model as operators. A recurrent neural network learns how to decompose these operators, such that the loss of the link prediction task is optimized [85]. *DRUM* [87] builds on the idea of *NeuralLP*. It increases the expressivity of *NeuralLP* by an improved representation of the differentiable operators. This allows for learning rules that cover a higher number of relations than *NeuralLP*. Furthermore, it uses a bidirectional recurrent neural network to optimize the operators. The bidirectional network capture's information about the backward and forward order in which the relations can appear in the rule [87]. Sadeghian et al. [87] show that *DRUM* outperforms state-of-the-art link prediction models like *MINERVA* [88] and *RotatE* [89] on the inductive link prediction task. *RuLES* [90] is also a neuro-symbolic method to rule mining. It learns clauses guided by a KG embedding. At each iteration, *RuLES* expands the existing clause such that the link prediction performance is optimized. The KG embedding provides the probability of a clause to be expanded by an atom [90]. That way, the search space for new clauses is effectively traversed, which increases the scalability and performance of *RuLES*. Ma et al. [91] induce explainable rules from KGs by creating a joint learning framework. The authors construct a rule-guided neural recommendation model, whereby inductive rules are mined from item-centric KGs. Other neuro-symbolic methods mine rules with the help of reinforcement learning [92]. First, an RL agent builds clauses with a curriculum learning strategy. Next, a rule mining system is built by applying the value function of the RL agent to efficiently guide the search for high-quality rules [92].

3.1.2. Pathfinding methods

The latter methods mine interpretable rules from KGs, which then are used to predict missing links between entities. This method is straightforward in discovering knowledge and building white-box AI models

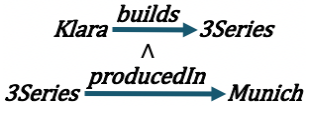
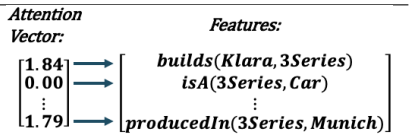
Model Interpretaion via		
Clause	Path	Attention Vector
$builds(x, y)$ $\wedge\ producedIn(y, z)$ $\implies\ worksIn(x, z)$		

Table 3: Three different types of explanations generated by interpretable models for the link prediction task "*Klara worksIn* ??" with the answer "*Munich*".

to predict missing links in the KG. Another line of research trains RL agents to find a path through KGs, the walked paths are the explanation of the prediction. A path-based interpretation is depicted in Table 3.

DeepPath [93] builds the bedrock of those methods. It uses *TransE* [22] to predict the tail for a head-relationship pair. Next, Xiong et al. [93] train a RL agent to find the most informative path linking the head and tail entity of the predicted triple. However, as the pathfinding is decoupled from the actual link prediction process, the faithfulness required for an explanation method is not given in the case of *DeepPath* [93]. The method inspired research on interpretable link prediction from the perspective of a pathfinding problem.

The first interpretable method deploying this perspective was *MINERVA* [88, 94]. *MINERVA* treats the KG vertices as states, and relations are the actions or paths to transition from one state to the other. A RL agent is trained to walk the optimal path through the KG to the correct tail entity. The policy network of the RL agent is an *LSTM* [95]. This allows the agent to consider the already walked path when choosing an action. The agent is rewarded whenever it walks to the correct tail entity. If the agent gets lost, it is not rewarded [94]. Lin et al. [96] observe that in an incomplete KG the RL agent receives low-quality rewards by false negatives in the training data. This reduces the generalization capacity of the agent at inference. In addition, *MINERVA* provides no optimal paths to the agent at training time. Thus, paths can be learned that accidentally lead to the correct answer. They address this problem by introducing a pre-trained one-hop embedding model, like *ComplEx* [97] or *ConvE* [98] to estimate the reward of unobserved facts. Lin et al. [96] show that this reduces the impact of false negatives in the training data.Furthermore, Lin et al. [96] force the agent to explore a diverse set of paths by applying randomly generated edge masks to the KG environment at training time. Hou et al. [99] are motivated by the same issues addressed by Lin et al. [96]. They adapt the search strategy of the RL agent. While in training, the agent selects paths guided by high-quality rules from a rule pool [99]. This results in less low-quality rewards and mitigates the impact of spurious paths. Bhowmik and de Melo [100] extend the path-finding method to the inductive link prediction domain. This allows link prediction for previously unseen entities. They achieve this by using a graph attention network for policy selection of the RL agent and a pre-trained *ConvE* [98] embedding as a reward estimator [100]. Furthermore, the interpretable pathfinding method on KGs is employed in various recommender systems [101, 102, 41]. In particular, *LOGER* [103] deploys a *Markov Logic Network* [104] to guide the RL agent to the optimal result. The *Markov Logic Network* learns personalized importance scores over paths from the user to recommendation items. To do so, the *Markov Logic Network* learns from previous user behaviour. Next, a RL agent finds the optimal path from the user to the recommendation item by optimizing the correctness of the recommendation and the importance scores of a path [103].

3.1.3. Embedding methods

Standard embedding methods show high predictive performance. However, they are not interpretable. A recent line of research tries to mitigate this issue by designing interpretable embedding models.

Xie et al. [105] proposes *ITransF*. *ITransF* consists of a translational embedding method similar to *STransE* [106]. In addition, they learn sparse attention vectors. The sparse attention vectors reflect how important a concept dimension is for the classification, and thus help to understand which underlying concepts different relations share [105]. An attention-based interpretation is depicted in Table 3. Attention vectors are not the only way the literature explores to achieve interpretability in sub-symbolic methods. Anelli et al. [107] introduces an interpretable recommender method relying on an FM. The latent factors of the FM are initialized by semantic features coming from a KG [107]. In addition, semantic features are injected into the learning process. Anelli et al. [107] show that this guarantees the interpretability of the methode's recommendations. *CrossE* [108] is a sub-symbolic embedding for link prediction that ex-

plains by analogy. *CrossE* predicts missing links by a translational model. Next, *CrossE* generates explanations of the predicted link. It does so by finding closed paths from the head to the predicted tail entity. Next, it searches for similar structured paths in the KG between entities already connected by the predicted relationship [108]. The found paths are shown to the user to justify the models' predicted link [108]. Another embedding-based method is called *INK* [65]. *INK* is used to create embeddings of vertices given their neighborhood. The embeddings may be used in downstream ML models for vertex-level classification [65]. *INK* learns a binary feature representation of the vertex's neighborhood for every vertex. The binary features allow users to interpret and understand the embeddings create by *INK* [65]. Steenwinckel et al. [65] show that *INK* performance is similar to the state-of-the-art black box methods, like *R-GCN* [109] and *RDF2Vec* [110].

TEM is neuro-symbolic method, relying on the attention vectors to establish interpretability [10]. First, *TEM* uses a decision tree to learn decision rules. Next, they design an attention network-based embedding model that uses the decision rules as features. The combination of the decision tree and the attention network guarantees that the recommendation process is fully interpretable [10]. Ai et al. [111] propose a different neuro-symbolic embedding method. The authors use at the core of their method a translation function similar to *TransE* [22]. The embedding is used to recommend items [111]. Next, the model generates an explanation [111]. First, possible closed paths between the head and tail entity are generated by breadth-first search. Then, the probability for each path is calculated by the translational function. The path with the highest probability is shown to a user to explain the predicted link [111]. Notably, Ruschel et al. [112] propose a similar method. However, they focus on the link prediction task instead of the recommendation task and use *SFE* [113, 114], a performance version of *PRA* [71], to mine rules used as explanations. Polleti et al. [115] also combine a rule mining and embedding method to compute interpretable recommendations. However, they add counterfactuals to their explanation. Thus, they do not simply justify why the user should trust the recommendation, but also why the user should not trust it. They do so by implementing *Snedegar's* theory of reasoning. For example, they select reasons, i.e. paths through the KG, against a recommendation. They do so by searching for reasons in favor of a competing recommendation that the initial recommendation is not satisfying [115].

Titles related to and/or discussed in this section are grouped regarding the survey's taxonomy in Table 4. The following section completes the taxonomy in regard to XAI on KGs.

3.2. Explainable Artificial Intelligence on Knowledge Graphs

Recalling that XAI is the mapping of the output of a ML model to its corresponding input (cf. section 2.1), this section provides an overview of those methods. Again, the following sections are structured based on the lines of research (cf. Figure 6) from our taxonomy for XAI on KGs.

3.2.1. Rule-based learning methods

Typically, RBL applications identify patterns by rules induced from the underlying data. Donadello and Dragoni [116], create the framework *SeXAI*, where human-understandable First Order Logic (FOL) formulas are produced. These formulas consist of features connected to the classes of a knowledge base. Furthermore, their connection to both the black box model and the annotations of the dataset allows for the explanation of the neural network predictions via semantic features. This renders the method neural-symbolic.[5] Other FOL based methods exist, such as Dervakos et al. [117], who uses the symbolic representation of a KG to express FOL rules in the terminology of the KG. Furthermore, *ExCut* [118] lives on the edge between RBL and GC. Gad-Elrab et al. [118] address the issue of latent embeddings. The authors show that latent embeddings have no significant contribution to the explainability of a prediction. Furthermore, their method creates human-understandable labels for clusters by applying rule mining methods on KG embeddings. The learned rules are used to generate labels for clusters, formed by an graph clustering method. The clustering method is embedding-based and clusters entities of a graph and is another contribution of the paper. Eventually, those labels are formed by combining entity relations according to the so-called cluster explanation rules.

3.2.2. Decomposition methods

A popular method to measure the importance of input features of a black box model is to decompose its predictions into separate terms. The partition of the prediction score of each term is then interpreted as the importance measure of the respective input feature. Recently, decomposition methods on Graph Neural Networks were proposed by Schnake et al. [119], who elab-

[5] FOL quantifies individual objects as variables, while higher-order logic also quantifies sets of individual objects.

Line of research	References	Representation	Task	Foundation
Rule Mining Methods Mining human-understandable clauses by optimizing hypothesis space travesel via effective heuristics and efficient search strategies.	*PRA* [71] *SWARM* [76] *RLvLR* [77] *RDF2Rules* [78] *AIME+* [72] *OP* [79] *ScaLeKB* [80] *AnyBURL* [81, 82] *SAFRAN* [83]	*SB*	LP	RBL
Mining human-understandable clauses with the help of differentiable operators.	*NeuralLP* [85] *DRUM* [87]	*NSB*	LP	RBL
Mining human-understandable clauses guided by a KG embedding.	*RuLES* [90]			
Mining human-understandable clauses via rule-guided neural recommendation model.	*RULEREC* [91]	*NSB*	R	RBL
Mining human-understandable clauses by guiding the rule search via a RL agent.	Reinforce Rule Miner [92]	*NSB*	LP	RL
Pathfinding Methods Finding human-understandable paths through the knowledge graph.	*MINERVA* [88] *Multi-hop* [96] Inductive Pathfinding [100] *RARL* [99]	*NSB*	LP	RL
	PGPR [101] *EKAR* [102] Distiling Knowledge [41] *LOGER* [103]	*NSB*	R	RL
Embedding Methods Generating semantic features from the KG as interpretations.	Feeding KG [107]	*SSB*	R	FM
Generating attention mask as interpretations.	*ITransF* [105],	*SSB*	LP	TL
Interpreting LP by analogies found in the KG.	*CrossE* [108]			
Interpretability via Decision Tree and Attention network.	*TEM* [10]	*NSB*	R	*O*
Interpretable binary neighborhood representations.	*INK* [65]	*SSB*	NC	TL
Generating and selecting rules via SFE and embedding function.	*XKE-e* [112]	*NSB*	LP	TL
Generating and selecting rules via BFS and embedding function.	Heterogeneous KGE [111]	*NSB*	R	TL
Generating reasons for and against a recommendation.	Snedegar Reasoning [115]			

Table 4: Literature review table of IML on KGs. Each item is classified by representation, task, foundational method, and line of research. Sometimes, approaches are constructed by multiple publications building upon each other. In this case, all related publications are cited.

orated on high-order explanations of Graph Neural Networks via so-called relevant walks. The authors man-

aged to identify groups of edges that commonly contribute to the GNN prediction by using a nested attribution scheme. At each step of this scheme, layer-wise relevance propagation (LRP) [120, 121] is applied in order to determine relevance. In accordance with our definition of XAI, the GNN LRP method explains any GNN prediction by extracting the most relevant paths from the input to the output of the GNN. Here, paths correspond to walks on the input graph U. The method is based on high-order *Taylor* expansions of the GNN output as well as chain-rules that are used to extract the *Taylor* expansion's terms [120]. Eventually, the GNN LRP method is transferable to large, nonlinear models. The method is applicable to a vast range of Graph Neural Network architectures and is set for application in sentiment analysis of text data, structure-property relationships in quantum chemistry, and image classification [121]. LRP remains one of the most popular ANN explanation methods, however, a detailed description of the GNN LRP method would exceed the scope of this survey. Figure 7 summarizes the proceeding. Furthermore, Ying et al. [122] introduce *GNNExplainer*, which highlights those modules of the GNN that explain the prediction. The authors determine these modules by learning the most suitable explanations via graph perturbation. The main idea is to perturb the input graph by adding or removing edges, nodes, or features in order to observe how these changes affect the Graph Neural Networks output. This perturbation involves making small changes to the graph's nodes and edges while keeping the rest of the structure intact. The technique is based on the assumption that the gradient of the prediction is known and the underlying data is independent and identically distributed. Hence, it generates an explanation for $\hat{y}$ as (G_S, X_S^F), where G_S is a sub-graph of the computation graph and X_S^F are the most important features for $\hat{y}$. Thereby, a vital component of *GNNExplainer* is learning real-valued graph- and feature-masks M, which is universal for all graph-based ML tasks.

3.2.3. Surrogate methods

Approximating the predictions of a black box model by deploying a more simple yet interpretable model is referred to as a surrogate method. The surrogate model is a simpler and interpretable model that is easier to understand than the original model. Zhang et al. [11] addresses the underlying assumption *GNNExplainer* to have access to the gradients of a model to learn explanations. The authors introduce *RelEx*, a method that works with relation extraction that can handle complex and cross-sentence relations. The authors lever both syntactic and semantic information to extract relations from text. Their method remains model-agnostic and does not rely on the iid assumption. Most importantly, *RelEx* resolves the limitation to differentiable relational models and therefore enhances practicality, compared to *GNNExplainer*.

Another representative from the field of model-agnostic GNN explanation methods is Vu and Thai [123]. Referred to as *Probabilistic Graphical Model-explainer*, the authors' manage to identify pivotal graph components and to deduce explanations in the form of a graphical model and conditional probabilities. Thereby, *Probabilistic Graphical Model-explainer* use graph-based representation to encode complex distributions over a multidimensional space. Compared to *GNNExplainer*, *Probabilistic Graphical Model-explainer* would provide an explanation in the form of a *Bayesian* network that estimates the probability that a vertex E has the predicted role given a realization of other vertices.

Probabilistic Graphical Model-explainer is, as well, model-agnostic and thus, can be applied to any graph-based AI model, independent of its architecture. This epitomizes a crucial advantage, especially when it comes to practical application. Furthermore, the model is applicable to other graphical models such as dependency networks or *Markov* networks. One may argue that local explanations, such as provided by Vu and Thai [123], lack generalizability for the whole graph. *PG-Explainer*, by Luo et al. [124], addresses this issue by parameterizing the process of explanation-generation, driven by the idea that neural network parameters are the same throughout the population. The intuition of the idea stems from conventional forward propagation-based methods and tries to identify a subgraph that contains explanatory vertices.

Other methods stress the proximity of a vertex of a KG, whose prediction is to be explained. In 2020, Huang et al. [125] proposed *GraphLIME* to explain any vertex by a set of features. Again, the method perturbs the input graph data around a specific instance of interest to create a modified graph. Next, the authors define a set of neighbours of a vertex, by the vertices whose distance from the is within a finite number of steps, frequently referred to as hops. Eventually, the nonlinear, feature-wise kernelized *Hilbert-Schmidt* Independence Criterion (HSIC) Lasso method is used to determine the least redundant feature vectors. This method is a supervised feature selection method and can be revised in Yamada et al. [126]. Essentially, the framework proposed by the authors learns a non-linear interpretable model in the local proximity of a predicted vertex. As opposed to *LIME*, a linear explanation model by Ribeiro et al. [49], *GraphLIME* takes the graph topology into

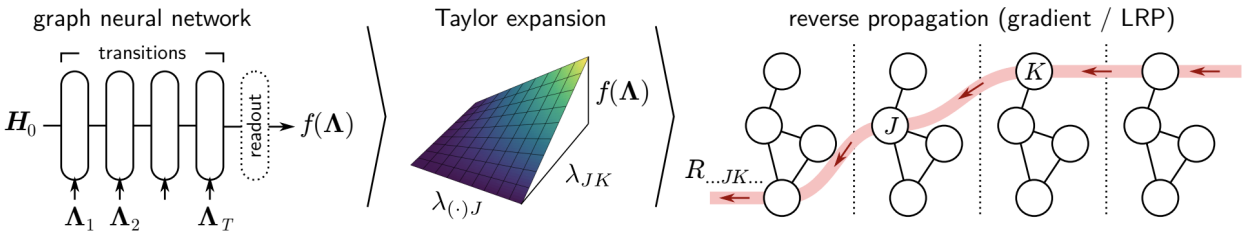
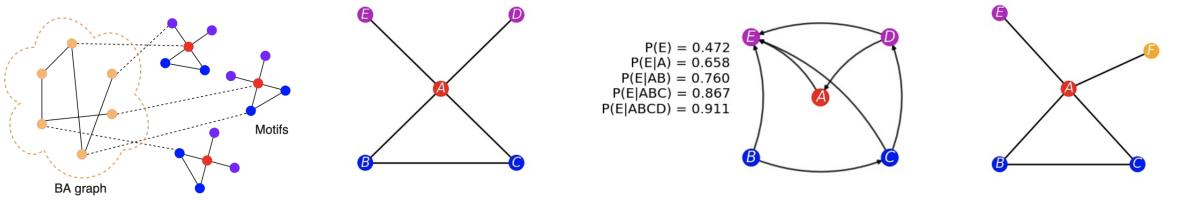


Figure 7: Exemplary GNN LRP method, taken from [121].



(a) Input graph. (b) Motif containing E. (c) PGM-Explainer. (d) GNNExplainer.

Figure 8: Exemplary *Probabilistic Graphical Model-explainer* framework compared to *GNNExplainer*, taken from [123].

account. Experimenting with the real-world datasets *CORA* and *Pubmed*, the explanations of *GraphLIME* are depicted as more descriptive, when compared to *GNNExplainer*.

Figure 9 displays the generic components of *GraphLIME*, with a vertex in red, whose prediction is to be explained, vertices in green, which are in the proximity of the vertex that is to be explained (a). From the feature vectors in red, the explanatory parts are determined (b), and presented to the user (c). On the vertex-level, *GraphSVX* by Duval and Malliaros [127] constructs a surrogate model on a perturbed dataset. Then, *GraphSVX* uses the surrogate model to compute the marginal contribution of vertices and features using the surrogate model binary masks (M_N, M_F). Eventually, *GraphSVX* draws on the so-called *Shapely* value, a concept from game theory, which represents the fair distribution of gains in a game. Put differently, *Shapely* values are a metric of the marginal contribution of each feature of a model's prediction. The authors approximate this metric in order to create a signification of an explanation obtained by *GraphSVX*. Additionally, the authors describe a unified framework of GNN explanations, consisting of a mask generator, a graph generator and an explanation generator, in which they locate competing models, such as *GNNExplainer*, *PGExplainer*, *PGM-Explainer* or *GraphLIME*. Even though the authors use a super-ordinate graph for generating explanations, the scope of *GraphSVX* is limited to vertex level predictions. *GraphSHAP* however, introduced by Perotti et al. [128] jointly makes use of the *Shapely* value, but on a GC level. The system decomposes the graph into motifs, whereby motifs are recurring patterns in the KG. For instance, a motif could be a triangle, a four-clique, or a cycle. Thus, *GraphSHAP* uses *Shapely* values to calculate the contribution of each motif to the graph classifier's output. Similarly, Schlichtkrull et al. [129] learn a straightforward classifier that predicts if an edge can be neglected or not, given its contribution to the prediction. For every node in every layer of a given, trained GNN, the system learns a classifier that predicts if that edge can be dropped. The authors claim to train the classifier in a fully differentiable fashion, employing stochastic gates and encouraging sparsity through the expected L_0 norm, which makes the classifier also interpretable attribution method to analyze Graph Neural Networks.

3.2.4. Graph generation methods

Some methods focus on training a graph generator instead of optimizing the input KG directly. For GC tasks, Yuan et al. [130] create a *XGNN*, a method to explain GNN predictions on the model-level. The authors train a separate graph generator in a RL setting that predicts a suitable way to add a new edge to the input graph. The generator produces a perturbation of the input graph, whereby changes are made in a way that af-

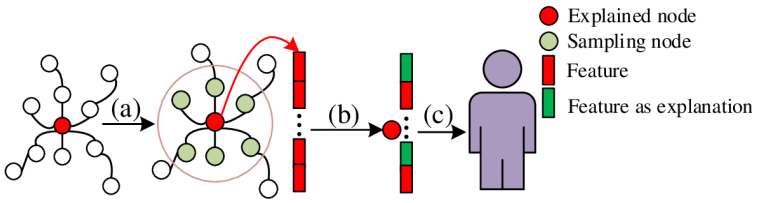


Figure 9: Generic components of *GraphLIME*, taken from Huang et al. [125]. The system samples neighbors of the node that is to be explained (in red) and vertices in green, which are in the proximity of the vertex that is to be explained (a). From the feature vectors in red, the explanatory parts are determined (b), and presented to the user (c).

fects the GNN's prediction. The RL policy works with manually generate graph rules that eventually explain the contributions of both nodes and edges. *XGNN* is appropriate for both graphs and KGs. Yet another model-agnostic but symbolic post-hoc explanation method for graph classifiers was proposed by Abrate and Bonchi [131] in 2021. The authors construct a counterfactual graph, which can, compared to *Shapely* values, provide a deterministic explanation of why a classifier made a prediction. The method proposed by the authors is a heuristic search scheme that can be conducted in two variants: (1) Oblivious, the algorithm searches for the optimal counterfactual graph by querying the black-box graph classifier and abstracting the original graph. (2) Data-driven, the algorithm gains additional access to a subset, superset or the original training dataset of the black box classifier. Similar to *SHAP* [50] and *LIME* [49], the authors rank statistics about the input data in order to achieve explainability. However, they do so by creating numerous counterfactual graphs and eventually stressing the most frequent edges. Thus, the method of Abrate and Bonchi [131] creates counterfactual graphs for different input data as an explanation, resulting in a global and model-agnostic XAI method. The authors forgo on integrating higher-order structures or graph properties into their method. A major issue for XAI in KG-based models is the occurrence of cyclic relations in the KG. Cycles occur when edges relate vertices in contradictory hierarchies. The main issue with a KG cycle is that reasoning over the graph in the form of choosing parent concepts within the KG hierarchy for generating explanations is rendered nondeterministic [132]. Sarker et al. [132] address the issue of cycles within a KG by providing a circle-free version of the existing *Wikipedia* KG (*WKG*). In the *WKG*, entities are articles in the English *Wikipedia* and relations are hyperlinks between them. The authors also introduce a type system, relying on the *Wikipedia* categories. Using breadth-first search, the authors break cyclic relations within the KG and thus create the so-called *Wikipedia* KG. Interestingly, the authors introduce a measure for the XAI-related quality of their KG, using an ILP-based metric, focusing on string similarity, which shall not be detailed here but can be revised in the original paper ([132]) in conjunction with Levenshtein [133]. Thus, explanations are tried to be extracted from existing KGs with cyclic relations, such as *DBpedia*. Eventually, the *WKG* can be used to identify the entities and types that are most relevant to a particular prediction and also to generate counterfactual explanations. Finally, the authors stipulate higher efficiency whilst minimal information loss by breaking cyclic relations. Other methods based on the exploitation of semantic information exist. Dragoni and Donadello [134] for instance, constructs a logical reasoning flow, referred to as explanation graph. The graph fulfills the formal criteria of a KG, as defined in Section 2, but differs from previously described methods since it maps concepts used by users to numeric features of a black box model. In the end, the graph is a representation of the data that was used by an AI model for its prediction, being matched to given KG concepts. In fact, the authors aim at rendering an explanation in logical language format. For that, they create a framework being able to process semantic information from a knowledge base, referred to as semantic features A_i, based on the work of [135]. In brief, semantic features represent the common attributes of an object, such as $3-Series \hookleftarrow (Car, BMW)$ and essentially are one kind of predicates of a FOL expression. The other kind of predicate are the output symbols of a black box model O_i that are to be rendered comprehensible. Hence, being graph-based, the authors define their system as consisting of unary predicates O_i, A_i as vertices and the logic relations between those predicates, coming from an AI system, as edges. The explanation graph hence combines singular knowledge fragments into a semantic reasoning system and thus creates explainability. Betz et al. [136] propose a method based on adversarial attacks on KG embedding models. Adversarial attacks involve the perturbation of data during training, which results in

model malfunction during testing. The authors leverage an efficient rule-learning technique, employing abductive reasoning to uncover triples that serve as logical justifications for specific predictions. Subsequently, the proposed attack centers around a straightforward concept: modifying or suppressing a triple within the most confident explanation. Remarkably, despite being independent of the model itself and requiring solely access to training data, the authors report outcomes on par with cutting-edge white-box attack techniques, which necessitate complete access to model architecture, learned embeddings, and loss functions. This surprising finding suggests that symbolic methods can partially retroactively elucidate KG embedding models.
Lastly, *approxSemanticCrossE*, presented by d'Amato et al. [137] uses semantic similarity to identify the most important entities and relations that contributed to the prediction. The technique employs semantic similarity to discern the pivotal entities and relationships influencing predictions. *approxSemanticCrossE* entails a three-step procedure: (1) Recognition of similar entities (i.e. the system identifies entities bearing the greatest resemblance to those involved in the link prediction, utilizing a semantic similarity metric). (2) Detection of similar relationships (i.e. *approxSemanticCrossE* pinpoints relationships that closely resemble the predicted link, employing a semantic similarity measure). (3) Exposition generation: The final step encompasses generating an exposition, which manifests as a compilation of the most similar entities and relations, accompanied by their respective semantic similarity scores. After having outlined the state of the art of XAI on KGs, Table 5 displays the results of the literature review for this section.

Additionally, 10 combines the taxonomy provided by this survey, summarized in the table above, with the discrimination of IML and XAI on KGs. The figure shows a heatmap of literature reviewed in this survey, displaying blank spots in current literature. The following section shall conclude the literature review and provide an outlook for future research in the field of KGs for CAI.

4. Conclusion

So far, we contributed to the research fields of XAI and IML by introducing the parent concept of CAI as well as a taxonomy for existing research in the conjunction of CAI with KGs. Now, the comprehensive overview of the preceding section shall be concluded before future research directions are described.

This survey provided a brief history of CAI on KGs before describing the proceeding of the survey. Next, we taxonimized CAI on KGs by the representation (cf. Section 2.3.1), the task (cf. Section 2.3.2), the foundational method (cf. Section 2.3.3), and the kind of CAI (cf. Section 2.1), which we refer to as either IML or XAI. We provided a detailed notion of our scope in the methodology and definitions section. Additionally, we made a case for the necessity of CAI on KGs, defending the concept CAI, XAI and IML on KGs. We divided the main body of this survey into an IML and a XAI part, where we described 55 selected titles from our survey. We can conclude that a significant amount of research works with a sub-symbolic representation of KGs. Neural network-based methods are strongly represented in both XAI and IML on KGs. Meanwhile, advances using RL are made. Figure 10 summarizes our results. The following section shall provide the reader with future research directions of CAI on KGs and concludes this survey.

Outlook and future research directions. Our survey unveiled that few XAI methods tackle the link prediction tasks and, at the same time, IML methods occur less frequently for vertex and graph classification. Thus, exploring XAI methods for link prediction is a research frontier with significant opportunities. For example, how can the decision process of a black box embedding model be reconstructed?
In addition, most XAI methods discussed in this paper are not KG specific. They do not leverage the semantic information of the node labels, edge labels, edge directions, or underlying ontology. Future XAI on KGs should leverage this information to enrich the explanations, leading to significant insides in the black box models' behavior.
This survey also recognizes that the comparative evaluation of the XAI methods on KGs needs to be clarified. There is yet to be an established ground for comparing the performance of different XAI methods on KG. Questions like which XAI method reflects the models' behavior more accurately are regularly unanswered. Future research has to establish common evaluation grounds in the form of standardized evaluation metrics.
At the same time, IML methods occur less frequently in vertex and graph clustering. Thus, IML methods for graph clustering hold significant potential for future research, as they promise to make graph clustering more performance and trustworthy. For example, an interpretable rule-mining method could learn rules from the KG structure that cluster similar KGs in the same group. Similarly, vertex clustering algorithms could be assessed and enriched with embedding-based IML methods, such as *CrossE* [137].

Line of research	References	Representation	Task	Foundation
Rule-based Using symbolic representation to extract First Order Logic rules from KGs.	*SeXAI* [116]	SB	Other	RBL
Extracting FOL rules using the terminology of KGs.	Rule-based explanations [117]	SB	C	RBL
Applying rule mining methods on embedding-based KG clustering.	*ExCut* [118]	SSB	C	RBL
Decomposition Identifying groups of relevant edges to construct relevant walks.	GNN LRP [120, 121]	SSB	Other	ANN
Surrogate Perturbing the input graph under IID assumptions.	*GNNExplainer* [122] *RelEx* [11]	SSB	NC	ANN
Using rules to create (graphical) explanations from relevant KG components.	*Probabilistic Graphical Model-explainer* [123] *PGExplainer* [124]	SSB	NC,GC	ANN
Explaining KG vertices by a set of explanatory features.	*GraphLIME* [125]	SSB	NC	ANN
Using Shapely values or attribution.	*GraphSHAP* [128] *GraphSVX* [127] Diff. edge masking [129]	SSB	NC, GC	TL
Graph generation Generating graphs by reinforced input graph perturbation.	*XGNN* [130]	SB	R	RL
Generating graphs with counterfactuals and inductive logic.	Counterfactual graph [131] *WKG* [132]	SB	GC	Other
Generating graphs by matching concepts of a KG.	Explanation graph [134]	SB	Other	Other
Generating graphs / embeddings using GANs and semantic similarity.	Adversarial Explanations [136] *approxSemanticCrossE* [137]	SB	Other	TL

Table 5: Literature review table of XAI on KGs. Each item is classified by representation, task, foundational method, and line of research. Sometimes, approaches are constructed by multiple publications building upon each other. In this case, all related publications are cited.

Furthermore, we see significant potential in improving the communication of the IML model's interpretation. While they are white-box models, the interpretation of their decision-making process is seldom user-friendly and requires expert ML knowledge. This survey encourages future research on IML on KGs to provide user-friendly communication strategies for the model's interpretation via stakeholder-specific interfaces. For example, an interface could visualize the model's decision-making process by embedding it in a context provided by the knowledge graph for non-expert users.
AI applications are moving outside the safe walls of labs and are invading our daily lives. Let us use the semantics provided by KGs to make AI applications safer.

Acknowledgments

We received numerous e-mails with feedback on our survey and would like to thank all researchers that contributed with their comments and constructive criticism to this paper. We especially would like to thank Judith Wewerka for her in-depth feedback on our work.

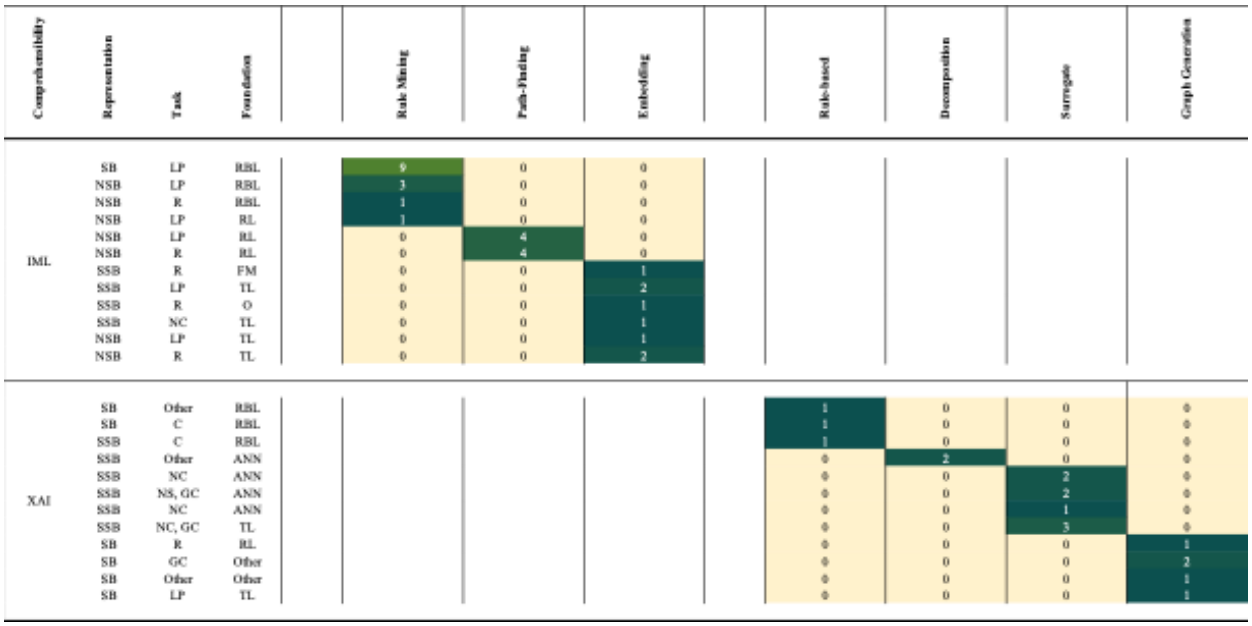


Figure 10: A heatmap of literature review results by representation, task and foundation. IML, XAI, and the determined lines of research are displayed separately.

References

[1] S. Bruckert, B. Finzel, U. Schmid, The next generation of medical decision support: A roadmap toward transparent expert companions, Frontiers in Artificial Intelligence 3 (2020). URL: https://www.frontiersin.org/article/10.3389/frai.2020.507973. doi:10.3389/frai.2020.507973.

[2] S. Schramm, M. Pieper, S. Vogl, Orthogonal procrustes based anomaly detection and error prediction for vehicle bills of materials, SSRN Electronic Journal (2022). doi:10.2139/ssrn.4120321.

[3] C. Wehner, M. Kertel, J. Wewerka, Interactive and intelligent root cause analysis in manufacturing with causal bayesian networks and knowledge graphs, in: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), 2023, pp. 1–7. doi:10.1109/VTC2023-Spring57618.2023.10199563.

[4] C. Wehner, F. Powlesland, B. Altakrouri, U. Schmid, Explainable online lane change predictions on a digital twin with a layer normalized lstm and layer-wise relevance propagation, in: H. Fujita, P. Fournier-Viger, M. Ali, Y. Wang (Eds.), Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence, Springer International Publishing, Cham, 2022, pp. 621–632. URL: https://doi.org/10.1007/978-3-031-08530-7_52. doi:10.1007/978-3-031-08530-7_52.

[5] The European Parliament and Council of European Union, Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, 2021. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206.

[6] G. Schwalbe, B. Finzel, A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts, Data Mining and Knowledge Discovery (2023). URL: https://doi.org/10.1007/s10618-022-00867-8. doi:10.1007/s10618-022-00867-8.

[7] G. Futia, A. Vetrò, On the integration of knowledge graphs into deep learning models for a more comprehensible ai—three challenges for future research, Information 11 (2020). URL: https://www.mdpi.com/2078-2489/11/2/122. doi:10.3390/info11020122.

[8] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215. URL: https://doi.org/10.1038/s42256-019-0048-x. doi:10.1038/s42256-019-0048-x.

[9] U. Schmid, Interactive learning with mutual explanations in relational domains, in: Human-Like Machine Intelligence, Oxford University Press, 2021, pp. 338–354. URL: https://doi.org/10.1093/oso/9780198862536.003.0017. doi:10.1093/oso/9780198862536.003.0017.

[10] X. Wang, X. He, F. Feng, L. Nie, T.-S. Chua, TEM: Tree-enhanced embedding model for explainable recommendation, in: P.-A. Champin, F. Gandon, M. Lalmas, P. G. Ipeirotis (Eds.), Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, ACM, 2018, pp. 1543–1552. URL: https://doi.org/10.1145/3178876.3186066. doi:10.1145/3178876.3186066.

[11] Y. Zhang, D. Defazio, A. Ramesh, RelEx: A model-agnostic relational model explainer, in: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, Association for Computing Machinery, 2021, pp. 1042–1049. URL: https://doi.org/10.1145/3461702.3462562. doi:10.1145/3461702.3462562.

[12] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid,

A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2022) 1–37. URL: https://doi.org/10.1145%2F3447772. doi:10.1145/3447772.

[13] M. Gaur, K. Faldu, A. Sheth, Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable?, IEEE Internet Computing 25 (2021) 51–59. doi:10.1109/MIC.2020.3031769.

[14] E. W. Schneider, Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis, Distributed by ERIC Clearinghouse, 1973. URL: https://eric.ed.gov/?id=ED088424.

[15] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (1995) 39–41. URL: https://doi.org/10.1145/219717.219748. doi:10.1145/219717.219748.

[16] S. Muggleton, Inductive logic programming, New Generation Computing 8 (1991) 295–318. URL: https://doi.org/10.1007/BF03037089. doi:10.1007/BF03037089.

[17] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of sparql, in: I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, L. M. Aroyo (Eds.), The Semantic Web - ISWC 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 30–43.

[18] J. R. Quinlan, Learning logical definitions from relations, Machine Learning 5 (1990) 239–266. URL: https://doi.org/10.1023/A:1022699322624. doi:10.1023/A:1022699322624.

[19] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), The Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 722–735.

[20] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 1247–1250. URL: https://doi.org/10.1145/1376616.1376746. doi:10.1145/1376616.1376746.

[21] A. Singhal, Introducing the knowledge graph: Things, not strings, 2012. URL: https://blog.google/products/search/introducing-knowledge-graph-things-not/.

[22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013, pp. 1–9. URL: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, Trans. Neur. Netw. 20 (2009) 61–80. URL: https://doi.org/10.1109/TNN.2008.2005605. doi:10.1109/TNN.2008.2005605.

[24] J. Webster, R. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS Quarterly 26 (2002). doi:10.2307/4132319.

[25] J. vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, A. Cleven, Reconstructing the giant: On the importance of rigour in documenting the literature search process., in: S. Newell, E. A. Whitley, N. Pouloudi, J. Wareham, L. Mathiassen (Eds.), ECIS, 2009, pp. 2206–2217. URL: http://dblp.uni-trier.de/db/conf/ecis/ecis2009.html#BrockeSNRPC09.

[26] S. R. Group, OECD, OECD and SCImago Research Group (CSIC) (2015), 2014.

[27] L. Bornmann, F. Moya Anegón, What proportion of excellent papers makes an institution one of the best worldwide? Specifying thresholds for the interpretation of the results of the SCImago Institutions Ranking and the Leiden Ranking., Journal of the Association for Information Science and Technology 65 (2014) 732–636.

[28] L. Bornmann, F. de Moya Anegón, L. Leydesdorff, The new Excellence Indicator in the World Report of the SCImago Institutions Rankings 2011, Journal of Informetrics 6 (2012) 333–335. doi:10.1016/j.joi.2011.11.006.

[29] G. Paré, M.-C. Trudel, M. Jaana, S. Kitsiou, Synthesizing information systems knowledge: A typology of literature reviews, Information & Management 52 (2015) 183–199. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378720614001116. doi:10.1016/j.im.2014.08.008.

[30] C. Molnar, Interpretable Machine Learning, 2 ed., 2022. URL: https://christophm.github.io/interpretable-ml-book.

[31] R. Roscher, B. Bohn, M. F. Duarte, J. Garcke, Explainable Machine Learning for Scientific Insights and Discoveries, IEEE Access 8 (2020) 42200–42216. doi:10.1109/ACCESS.2020.2976199.

[32] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proceedings of the National Academy of Sciences 116 (2019) 22071–22080. doi:10.1073/pnas.1900654116.

[33] Z. C. Lipton, The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery., Queue 16 (2018) 31–57. doi:10.1145/3236386.3241340.

[34] H. Xu, C. Jiang, X. Liang, L. Lin, Z. Li, Reasoning-RCNN: Unifying Adaptive Global Reasoning Into Large-Scale Object Detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019, pp. 6412–6421. doi:10.1109/CVPR.2019.00658.

[35] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: a survey, Artificial Intelligence 302 (2021). doi:10.1016/j.artint.2021.103627.

[36] F. Bianchi, G. Rossiello, L. Costabello, M. Palmonari, P. Minervini, Knowledge graph embeddings and explainable AI, in: Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, 2020, pp. 49–72. URL: https://doi.org/10.3233/SSW200011. doi:10.3233/SSW200011.

[37] F. Lecue, On the role of knowledge graphs in explainable ai, Semantic Web 11 (2020) 41–51. doi:10.3233/SW-190374.

[38] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, Found. Trends Inf. Retr. 14 (2020) 1–101. URL: https://doi.org/10.1561/1500000066. doi:10.1561/1500000066.

[39] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 1–19. doi:10.1109/TPAMI.2022.3204236.

[40] A. Seeliger, M. Pfaff, H. Krcmar, Semantic web technologies for explainable machine learning models: A literature review., PROFILES/SEMEX@ ISWC 2465 (2019) 1–16.

[41] Y. Zhang, X. Xu, H. Zhou, Y. Zhang, Distilling structured

knowledge into embeddings for explainable and accurate recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 735–743. URL: https://doi.org/10.1145/3336191.3371790. doi:10.1145/3336191.3371790.

[42] V. Lully, P. Laublet, M. Stankovic, F. Radulovic, Enhancing explanations in recommender systems with knowledge graphs, Procedia Computer Science 137 (2018) 211–222. URL: https://www.sciencedirect.com/science/article/pii/S1877050918316259. doi:https://doi.org/10.1016/j.procs.2018.09.020, proceedings of the 14th International Conference on Semantic Systems 10th – 13th of September 2018 Vienna, Austria.

[43] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160. doi:10.1109/ACCESS.2018.2870052.

[44] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery., Queue 16 (2018) 31–57. URL: https://doi.org/10.1145/3236386.3241340. doi:10.1145/3236386.3241340.

[45] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nature Communications 10 (2019). URL: https://doi.org/10.1038/s41467-019-08987-4. doi:10.1038/s41467-019-08987-4.

[46] O. Lahav, N. Mastronarde, M. van der Schaar, What is interpretable? using machine learning to design interpretable decision-support systems, CoRR abs/1811.10799 (2018). arXiv:1811.10799.

[47] Y. Zhou, S. Booth, M. T. Ribeiro, J. Shah, Do feature attribution methods correctly attribute features?, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 9623–9633. URL: https://ojs.aaai.org/index.php/AAAI/article/view/21196. doi:10.1609/aaai.v36i9.21196.

[48] Y. Zhai, M. Shah, Visual attention detection in video sequences using spatiotemporal cues, in: Proceedings of the 14th ACM International Conference on Multimedia, Association for Computing Machinery, 2006, pp. 815–824. URL: https://doi.org/10.1145/1180639.1180824. doi:10.1145/1180639.1180824.

[49] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should i trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 1135–1144. URL: https://doi.org/10.1145/2939672.2939778. doi:10.1145/2939672.2939778.

[50] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4768–4777. URL: https://dl.acm.org/doi/10.5555/3295222.3295230. doi:10.5555/3295222.3295230.

[51] S. Verma, J. P. Dickerson, K. E. Hines, Counterfactual explanations for machine learning: A review, ArXiv abs/2010.10596 (2020).

[52] J. Rabold, M. Siebers, U. Schmid, Explaining black-box classifiers with ilp – empowering lime with aleph to approximate non-linear decisions with relational rules, in: F. Riguzzi, E. Bellodi, R. Zese (Eds.), Inductive Logic Programming, Springer International Publishing, Cham, 2018, pp. 105–117. URL: https://doi.org/10.1007/978-3-319-99960-9_7. doi:10.1007/978-3-319-99960-9_7.

[53] B. Sanchez-Lengeling, E. Reif, A. Pearce, A. Wiltschko, A gentle introduction to graph neural networks, Distill 6 (2021). URL: https://distill.pub/2021/gnn-intro. doi:10.23915/distill.00033.

[54] F. Harary, R. Z. R. Z. Norman, D. Cartwright, Structural models: an introduction to the theory of directed graphs, Wiley, 1965.

[55] J. Gallian, A dynamic survey of graph labelingx, Electron. J. Combin., Dynamic Surveys 19 (2000).

[56] U. Schmid, V. Tresp, M. Bethge, K. Kersting, R. Stiefelhagen, Künstliche Intelligenz – Die dritte Welle, in: R. H. Reussner, A. Koziolek, R. Heinrich (Eds.), INFORMATIK 2020, Gesellschaft für Informatik, Bonn, 2021, pp. 91–95. doi:10.18420/inf2020_08.

[57] A. S. d'Avila Garcez, L. Lamb, Neurosymbolic ai: The 3rd wave, ArXiv abs/2012.05876 (2020).

[58] C. d'Amato, Machine learning for the semantic web: Lessons learnt and next research directions, Semantic Web 11 (2020) 195–203. doi:10.3233/SW-200388.

[59] H. Xiao, Y. Chen, X. Shi, Knowledge graph embedding based on multi-view clustering framework, IEEE Transactions on Knowledge and Data Engineering 33 (2021) 585–596. doi:10.1109/TKDE.2019.2931548.

[60] Z. Huang, Link prediction based on graph topology: The predictive value of generalized clustering coefficient, Econometrics: Applied Econometrics & Modeling eJournal (2010). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1634014. doi:10.2139/ssrn.1634014.

[61] F. Tian, B. Gao, Q. Cui, E. Chen, T.-Y. Liu, Learning deep representations for graph clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 28, 2014, pp. 1–7. URL: https://ojs.aaai.org/index.php/AAAI/article/view/8916. doi:10.1609/aaai.v28i1.8916.

[62] C. Schmitz, A. Hotho, R. Jäschke, G. Stumme, Content aggregation on knowledge bases using graph clustering, in: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 530–544.

[63] M. Elbattah, M. Roushdy, M. Aref, A.-B. M.Salem, Large-scale entity clustering based on structural similarity within knowledge graphs, Big Data Analytics: Tools and Technology for Effective Planning (2017) 311–334. doi:10.1201/b21822-14.

[64] J. Liu, L. Duan, A survey on knowledge graph-based recommender systems, in: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), volume 5, 2021, pp. 2450–2453. doi:10.1109/IAEAC50856.2021.9390863.

[65] B. Steenwinckel, G. Vandewiele, M. Weyns, T. Agozzino, F. D. Turck, F. Ongenae, Ink: knowledge graph embeddings for node classification, Data Mining and Knowledge Discovery 36 (2022) 620–667. URL: https://doi.org/10.1007/s10618-021-00806-z. doi:10.1007/s10618-021-00806-z.

[66] D. Hwang, S. Yang, Y. Kwon, K. H. Lee, G. Lee, H. Jo, S. Yoon, S. Ryu, Comprehensive study on molecular supervised learning with graph neural networks, Journal of Chemical Information and Modeling 60 (2020) 5936–5945.

[67] J. B. Lee, R. Rossi, X. Kong, Graph classification using structural attention, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2018, pp.

1666–1674. URL: https://doi.org/10.1145/3219819.3219980. doi:10.1145/3219819.3219980.

[68] M. Ali, M. Berrendorf, C. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (2021) 1–1. doi:10.1109/TPAMI.2021.3124805.

[69] W. W. Cohen, C. D. Page, Jr., Polynomial learnability and inductive logic programming: Methods and results, New Generation Computing 13 (1995) 369–409.

[70] V. Svátek, J. Rauch, M. Ralbovský, Ontology-enhanced association mining, in: M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenič, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek, M. van Someren (Eds.), Semantics, Web and Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 163–179.

[71] N. Lao, T. Mitchell, W. W. Cohen, Random walk inference and learning in a large scale knowledge base, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 529–539. URL: https://aclanthology.org/D11-1049.

[72] L. Galárraga, C. Teflioudi, K. Hose, F. M. Suchanek, Fast rule mining in ontological knowledge bases with amie+, The VLDB Journal 24 (2015) 707–730. URL: https://doi.org/10.1007/s00778-015-0394-1. doi:10.1007/s00778-015-0394-1.

[73] D. Bokde, S. Girase, D. Mukhopadhyay, Matrix factorization model in collaborative filtering algorithms: A survey, Procedia Computer Science 49 (2015) 136–146.

[74] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, Cambridge, MA, USA, 2018.

[75] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, Omnipress, Madison, WI, USA, 2011, p. 809–816.

[76] M. Barati, Q. Bai, Q. Liu, SWARM: An approach for mining semantic association rules from semantic web data, in: R. Booth, M.-L. Zhang (Eds.), PRICAI 2016: Trends in Artificial Intelligence, Springer International Publishing, Cham, 2016, pp. 30–43. URL: https://doi.org/10.1007/978-3-319-42911-3_3. doi:10.1007/978-3-319-42911-3_3.

[77] P. G. Omran, K. Wang, Z. Wang, Scalable rule learning via learning representation, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18, AAAI Press, 2018, p. 2149–2155.

[78] Z. Wang, J.-Z. Li, Rdf2rules: Learning rules from rdf knowledge bases by mining frequent predicate cycles, CoRR abs/1512.07734 (2015). URL: http://arxiv.org/abs/1512.07734.

[79] Y. Chen, S. Goldberg, D. Z. Wang, S. S. Johri, Ontological pathfinding, in: Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 835–846. URL: https://doi.org/10.1145/2882903.2882954. doi:10.1145/2882903.2882954.

[80] Y. Chen, D. Z. Wang, S. Goldberg, ScaLeKB: Scalable learning and inference over large knowledge bases, The VLDB Journal 25 (2016) 893–918. URL: https://doi.org/10.1007/s00778-016-0444-3. doi:10.1007/s00778-016-0444-3.

[81] C. Meilicke, M. W. Chekol, D. Ruffinelli, H. Stuckenschmidt, Anytime bottom-up rule learning for knowledge graph completion, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3137–3143. URL: https://doi.org/10.24963/ijcai.2019/435. doi:10.24963/ijcai.2019/435.

[82] C. Meilicke, M. W. Chekol, M. Fink, H. Stuckenschmidt, Reinforced anytime bottom up rule learning for knowledge graph completion, ArXiv abs/2004.04412 (2020).

[83] S. Ott, C. Meilicke, M. Samwald, SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models, in: 3rd Conference on Automated Knowledge Base Construction, 2021, pp. 1–18. URL: https://openreview.net/forum?id=jCt9S_3w_S9.

[84] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Trans. Knowl. Discov. Data 15 (2021). URL: https://doi.org/10.1145/3424672. doi:10.1145/3424672.

[85] F. Yang, Z. Yang, W. W. Cohen, Differentiable learning of logical rules for knowledge base reasoning, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017, pp. 1–11. URL: https://proceedings.neurips.cc/paper/2017/file/0e55666a4ad822e0e34299df3591d979-Paper.pdf.

[86] W. W. Cohen, F. Yang, K. Mazaitis, Tensorlog: A probabilistic database implemented using deep-learning infrastructure, Journal of Artificial Intelligence Research 67 (2020) 285–325. URL: https://doi.org/10.1613/jair.1.11944. doi:10.1613/jair.1.11944.

[87] A. Sadeghian, M. Armandpour, P. Ding, D. Z. Wang, Drum: End-to-end differentiable rule mining on knowledge graphs, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 1–11. URL: https://dl.acm.org/doi/10.5555/3454287.3455662. doi:10.5555/3454287.3455662.

[88] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, A. McCallum, Go for a walk and arrive at the answer: Reasoning over knowledge bases with reinforcement learning, in: 6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS, 2017, pp. 1–18.

[89] Z. Sun, Z. Deng, J. Nie, J. Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019, pp. 1–18. URL: https://openreview.net/forum?id=HkgEQnRqYQ.

[90] V. T. Ho, D. Stepanova, M. H. Gad-Elrab, E. Kharlamov, G. Weikum, Rule learning from knowledge graphs guided by embedding models, in: D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee, E. Simperl (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham, 2018, pp. 72–90. URL: https://doi.org/10.1007/978-3-030-00671-6_5. doi:10.1007/978-3-030-00671-6_5.

[91] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, X. Ren, Jointly learning explainable rules for recommendation with knowledge graph, in: The World Wide Web Conference, Association for Computing Machinery, 2019, pp. 1210–1221. URL: https://doi.org/10.1145/3308558.3313607. doi:10.1145/3308558.3313607.

[92] L. Chen, S. Jiang, J. Liu, C. Wang, S. Zhang, C. Xie, J. Liang, Y. Xiao, R. Song, Rule mining over knowledge graphs via reinforcement learning, Knowledge-Based Systems 242 (2022) 108371. URL: https://www.sciencedirect.com/science/article/pii/S095070512200140X. doi:https://doi.org/10.1016/j.knosys.2022.108371.

[93] W. Xiong, T. Hoang, W. Y. Wang, Deeppath: A reinforcement learning method for knowledge graph reasoning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017, pp. 564–573. URL: https://aclanthology.org/D17-1060. doi:10.18653/v1/D17-1060.

[94] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, A. McCallum, Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018, pp. 1–18. URL: https://openreview.net/forum?id=Syg-YfWCW.

[95] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735.

[96] X. V. Lin, R. Socher, C. Xiong, Multi-hop knowledge graph reasoning with reward shaping, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018, pp. 3243–3253. URL: https://aclanthology.org/D18-1362. doi:10.18653/v1/D18-1362.

[97] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 2071–2080.

[98] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018, pp. 1–9. URL: https://dl.acm.org/doi/10.5555/3504035.3504256. doi:10.5555/3504035.3504256.

[99] Z. Hou, X. Jin, Z. Li, L. Bai, Rule-aware reinforcement learning for knowledge graph reasoning, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 4687–4692. URL: https://aclanthology.org/2021.findings-acl.412.pdf.

[100] R. Bhowmik, G. de Melo, Explainable link prediction for emerging entities in knowledge graphs, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham, 2020, pp. 39–55. URL: https://doi.org/10.1007/978-3-030-62419-4_3. doi:10.1007/978-3-030-62419-4_3.

[101] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, Association for Computing Machinery, New York, NY, USA, 2019, p. 285–294. URL: https://doi.org/10.1145/3331184.3331203. doi:10.1145/3331184.3331203.

[102] W. Song, Z. Duan, Z. Yang, H. Zhu, M. Zhang, J. Tang, Ekar: Explainable knowledge graph-based recommendation via deep reinforcement learning, ArXiv abs/1906.09506 (2019).

[103] Y. Zhu, Y. Xian, Z. Fu, G. de Melo, Y. Zhang, Faithfully explainable recommendation via neural logic reasoning, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 3083–3090. URL: https://aclanthology.org/2021.naacl-main.245. doi:10.18653/v1/2021.naacl-main.245.

[104] M. Qu, J. Tang, Probabilistic logic neural networks for reasoning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in neural information processing systems, volume 32, Curran Associates, Inc., 2019, pp. 1–13. URL: https://proceedings.neurips.cc/paper/2019/file/13e5ebb0fa112fe1b31a1067962d74a7-Paper.pdf.

[105] Q. Xie, X. Ma, Z. Dai, E. Hovy, An interpretable knowledge transfer model for knowledge base completion, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 950–962. URL: https://aclanthology.org/P17-1088. doi:10.18653/v1/P17-1088.

[106] D. Q. Nguyen, K. Sirts, L. Qu, M. Johnson, STransE: a novel embedding model of entities and relationships in knowledge bases, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2016, pp. 460–466. URL: https://aclanthology.org/N16-1054. doi:10.18653/v1/N16-1054.

[107] V. W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone, J. Trotta, How to Make Latent Factors Interpretable by Feeding Factorization Machines with Knowledge Graphs, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, volume 11778, Springer International Publishing, Cham, 2019, pp. 38–56. doi:10.1007/978-3-030-30793-6_3.

[108] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, H. Chen, Interaction embeddings for prediction and explanation in knowledge graphs, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Association for Computing Machinery, 2019, pp. 96–104. URL: https://doi.org/10.1145/3289600.3291014. doi:10.1145/3289600.3291014.

[109] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European semantic web conference, 2018, pp. 593–607.

[110] P. Ristoski, H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I, Springer-Verlag, 2016, pp. 498–514. URL: https://doi.org/10.1007/978-3-319-46523-4_30. doi:10.1007/978-3-319-46523-4_30.

[111] Q. Ai, V. Azizi, X. Chen, Y. Zhang, Learning heterogeneous knowledge base embeddings for explainable recommendation, Algorithms 11 (2018). URL: https://www.mdpi.com/1999-4893/11/9/137. doi:10.3390/a11090137.

[112] A. Ruschel, A. C. Gusmão, G. P. Polleti, F. G. Cozman, Explaining completions produced by embeddings of

knowledge graphs, in: G. Kern-Isberner, Z. Ognjanović (Eds.), European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer International Publishing, Cham, 2019, pp. 324–335. URL: https://doi.org/10.1007/978-3-030-29765-7_27. doi:10.1007/978-3-030-29765-7_27.

[113] M. Gardner, T. M. Mitchell, Efficient and expressive knowledge base completion using subgraph feature extraction, in: EMNLP, 2015, pp. 1–9.

[114] A. C. Gusmão, A. H. C. Correia, G. D. Bona, F. G. Cozman, Interpreting embedding models of knowledge bases: A pedagogical approach, ICML Workshop on Human Interpretability in Machine Learning (WHI). Stockholm, Sweden (2018).

[115] G. P. Polleti, D. L. de Souza, F. G. Cozman, Why should i not follow you? reasons for and reasons against in responsible recommender systems, CoRR abs/2009.01953 (2020).

[116] I. Donadello, M. Dragoni, SeXAI: A semantic explainable artificial intelligence framework, in: M. Baldoni, S. Bandini (Eds.), AIxIA 2020 – Advances in Artificial Intelligence, Springer International Publishing, 2021, pp. 51–66. URL: https://doi.org/10.1007/978-3-030-77091-4_4. doi:10.1007/978-3-030-77091-4_4.

[117] E. Dervakos, O. Menis-Mastromichalakis, A. Chortaras, G. Stamou, Computing rule-based explanations of machine learning classifiers using knowledge graphs, ArXiv abs/2202.03971 (2022).

[118] M. H. Gad-Elrab, D. Stepanova, T.-K. Tran, H. Adel, G. Weikum, ExCut: Explainable embedding-based clustering over knowledge graphs, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, 2020, pp. 218–237. URL: https://doi.org/10.1007/978-3-030-62419-4_13. doi:10.1007/978-3-030-62419-4_13.

[119] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, G. Montavon, Xai for graphs: Explaining graph neural network predictions by identifying relevant walks, CoRR abs/2006.03589 (2020).

[120] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE 10 (2015) 1–46. URL: https://doi.org/10.1371/journal.pone.0130140. doi:10.1371/journal.pone.0130140.

[121] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, K.-R. Müller, Layer-Wise Relevance Propagation: An Overview, Springer International Publishing, Cham, 2019, pp. 193–209. URL: https://doi.org/10.1007/978-3-030-28954-6_10. doi:10.1007/978-3-030-28954-6_10.

[122] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, GNNExplainer: Generating explanations for graph neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019, p. 9244–9255. URL: https://dl.acm.org/doi/10.5555/3454287.3455116. doi:10.5555/3454287.3455116.

[123] M. Vu, M. T. Thai, PGM-Explainer: Probabilistic graphical model explanations for graph neural networks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 12225–12235. URL: https://dl.acm.org/doi/abs/10.5555/3495724.3496749. doi:10.5555/3495724.3496749.

[124] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, X. Zhang, Parameterized explainer for graph neural network, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020, p. 19620–19631. URL: https://dl.acm.org/doi/10.5555/3495724.3497370. doi:10.5555/3495724.3497370.

[125] Q. Huang, M. Yamada, Y. Tian, D. Singh, Y. Chang, GraphLIME: Local interpretable model explanations for graph neural networks, IEEE Transactions on Knowledge and Data Engineering (2022) 1–6. doi:10.1109/TKDE.2022.3187455.

[126] M. Yamada, J. Tang, J. Lugo-Martinez, E. Hodzic, R. Shrestha, A. Saha, H. Ouyang, D. Yin, H. Mamitsuka, C. Sahinalp, P. Radivojac, F. Menczer, Y. Chang, Ultra high-dimensional nonlinear feature selection for big biological data, IEEE Transactions on Knowledge and Data Engineering 30 (2018) 1352–1365. doi:10.1109/TKDE.2018.2789451.

[127] A. Duval, F. D. Malliaros, GraphSVX: Shapley value explanations for graph neural networks, in: N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, J. A. Lozano (Eds.), Machine Learning and Knowledge Discovery in Databases. Research Track, Springer International Publishing, Cham, 2021, pp. 302–318. URL: https://doi.org/10.1007/978-3-030-86520-7_19. doi:10.1007/978-3-030-86520-7_19.

[128] A. Perotti, P. Bajardi, F. Bonchi, A. Panisson, GRAPHSHAP: Motif-based explanations for black-box graph classifiers, CoRR abs/2202.08815 (2022).

[129] M. S. Schlichtkrull, N. D. Cao, I. Titov, Interpreting graph neural networks for NLP with differentiable edge masking, in: International Conference on Learning Representations, 2021, pp. 1–21. URL: https://openreview.net/forum?id=WznmQa42ZAx.

[130] H. Yuan, J. Tang, X. Hu, S. Ji, XGNN: Towards model-level explanations of graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 430–438. URL: https://doi.org/10.1145/3394486.3403085. doi:10.1145/3394486.3403085.

[131] C. Abrate, F. Bonchi, Counterfactual graphs for explainable classification of brain networks, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2021, pp. 2495–2504. URL: https://doi.org/10.1145/3447548.3467154. doi:10.1145/3447548.3467154.

[132] M. K. Sarker, J. Schwartz, P. Hitzler, L. Zhou, S. Nadella, B. Minnery, I. Juvina, M. L. Raymer, W. R. Aue, Wikipedia knowledge graph for explainable ai, in: B. Villazón-Terrazas, F. Ortiz-Rodríguez, S. M. Tiwari, S. K. Shandilya (Eds.), Knowledge Graphs and Semantic Web, Springer International Publishing, Cham, 2020, pp. 72–87. URL: https://doi.org/10.1007/978-3-030-65384-2_6. doi:10.1007/978-3-030-65384-2_6.

[133] V. I. Levenshtein, On the minimal redundancy of binary error-correcting codes, Inf. Control. 28 (1975) 268–291. URL: https://doi.org/10.1016/S0019-9958(75)90300-9. doi:10.1016/S0019-9958(75)90300-9.

[134] M. Dragoni, I. Donadello, A knowledge-based strategy for xai: The explanation graph, Semantic Web Journal (2022).

[135] D. Doran, S. Schulz, T. R. Besold, What does explainable ai really mean? a new conceptualization of perspectives, CEUR Workshop Proceedings 2071 (2018). URL: https://openaccess.city.ac.uk/id/eprint/18660/.

[136] P. Betz, C. Meilicke, H. Stuckenschmidt, Adversarial explanations for knowledge graph embeddings, in: L. D.

Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 2820–2826. URL: https://doi.org/10.24963/ijcai.2022/391. doi:10.24963/ijcai.2022/391.

[137] C. d'Amato, P. Masella, N. Fanizzi, An approach based on semantic similarity to explaining link predictions on knowledge graphs, in: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT '21, Association for Computing Machinery, New York, NY, USA, 2022, p. 170–177. URL: https://doi.org/10.1145/3486622.3493956. doi:10.1145/3486622.3493956.

Appendix A. Selected questionnaire answers.

Which papers do you think are crucial to cite in such a survey?
To the best of my knowledge, very few papers tackled the explainability of knowledge graphs. The explainability of graphs was tackled by several works. [...] Although these approaches generalise well to KG, I believe there is some space for more targeted approach, which leverage the different types of edges / atoms more explicitly into their functioning, proposing more balanced or type-specific explanations using these additional information.

Which major outlets (conferences/journals) have you seen published literature on Comprehensible Artificial Intelligence on Knowledge Graphs?
[Multiple answers, all of which were integrated into this survey.]

Are there topics you would like to see discussed in such a survey?
I think real-world comprehensibility entails a mixture of / a continuum between explanation of the mechanics of the model and explanation of the world. In non-trivial knowledge domains, explanations will likely include information that is novel to the user and that not only describes how the model worked, they will also need to be shown and explained the world-knowledge that is relevant, and need to judge the relevance and validity of that knowledge.

Where do you see future research directions for Comprehensible Artificial Intelligence on Knowledge Graphs?
I have the suspicion that approaches existing approaches [...] are close to optimal when it comes to purely graph-structured data. I think we will see increasing inclusion of additional, non-graph data (especially text data processed by language models) in order to go beyond the current state-of-the-art of predictive performance. [...]