Towards Explainable Automated Knowledge Engineering with Human-in-the-loop

Bohui Zhang [a,*], Albert Meroño-Peñuela [a] and Elena Simperl [a]
[a] *Department of Informatics, King's College London, London, United Kingdom*
E-mails: bohui.zhang@kcl.ac.uk, albert.merono@kcl.ac.uk, elena.simperl@kcl.ac.uk

Abstract. Knowledge graphs are important in human-centered AI as they provide large labeled machine learning datasets, enhance retrieval-augmented generation, and generate explanations. However, knowledge graph construction has evolved into a complex, semi-automatic process that increasingly relies on black-box deep learning models and heterogeneous data sources to scale. The knowledge graph lifecycle is not transparent, accountability is limited, and there are no accounts of, or indeed methods to determine, how fair a knowledge graph is in downstream applications. Knowledge graphs are thus at odds with AI regulation, for instance, the EU's AI Act, and with ongoing efforts elsewhere in AI to audit and debias data and algorithms. This paper reports on work towards designing explainable (XAI) knowledge-graph construction pipelines with humans in-the-loop and discusses research topics in this area. Our work is based on a systematic literature review, in which we study tasks in knowledge graph construction that are often automated, as well as common methods to explain how they work and their outcomes, and an interview study with 13 people from the knowledge engineering community. To analyze the related literature, we introduce use cases, their related goals for XAI methods in knowledge graph construction, and the gaps in each use case. To gain an understanding of the role of explainable models in practical scenarios, and reveal the requirements for improving the current XAI methods, we designed interview questions covering broad transparency and explainability topics, along with example discussion sessions using examples from the literature review. From practical knowledge engineering experience, we collect requirements for designing XAI methods, propose design blueprints, and outline directions for future research: (i) tasks in knowledge graph construction where manual input remains essential and where AI assistance could be beneficial; (ii) integrating XAI methods into established knowledge engineering practices to improve stakeholder experience; (iii) the need to evaluate how effective explanations genuinely are making human-machine collaboration in knowledge graph construction more trustworthy; (iv) adapting explanations for multiple use cases; and (v) verifying and applying the XAI design blueprint in practical settings.

Keywords: knowledge graph, knowledge graph construction, knowledge engineering, transparency, explainability, explainable AI, trustworthy AI

1. Introduction

To reach its potential, AI needs data and context. Without the right (amounts of) data, machine learning (ML) cannot identify patterns or make predictions. Without a deeper understanding of context, AI applications cannot engage people in a meaningful way. Knowledge graphs (KGs) [1, 2], a term coined by Google in 2012 to refer to its general-purpose knowledge base, are critical to both: they reduce the need for large labeled ML datasets [3], enhance pre-trained language models (PLMs) [4, 5], and generate explanations [6]. KGs are routinely used alongside ML in many applications, including search, question answering, recommendation [7] and, in industry contexts, enterprise data management, digital twins, supply chain management, procurement, and regulatory compliance [8]. Moreover,

*Corresponding author. E-mail: bohui.zhang@kcl.ac.uk.

1570-0844/$35.00 © 0 – IOS Press. All rights reserved.

with the rise of large language models (LLMs) such as GPT [9, 10] and Llama series [11, 12], KGs and LLMs have influenced each other in both ways: LLMs for KGs (using LLMs for KG construction and maintenance) and KGs for LLMs (using KGs to train, prompt, augment, and evaluate LLMs) [13–15].

As AI applications produce and consume more data, engineering KGs has evolved into a complex, semi-automatic process that increasingly relies on opaque deep-learning models and vast collections of heterogeneous data sources to scale to graphs with millions of entities and billions of statements [16–19]. The KG lifecycle is not transparent [20], accountability is limited, and accounts of how biased a KG is [21] or how fair the downstream applications that use it are patchy [22]. In recent works, KGs themselves are meant to make ML models explainable [6] and hence facilitate such compliance tasks, but that would imply that the KG lifecycle abides by the same rules.

We argue that this is not yet the case. As referred to in our previous work [23], questions regarding the user-centric aspects of knowledge engineering are not yet fully answered, such as users' tasks and goals, the way that they interact with KGs, KG construction (KGC) tools, and KG-related applications [24]. Up-to-date comparative surveys regarding the scale, complexity, and degree of automation of KG construction systems nowadays are needed. User-centric design and empirical methods should be established for transparent KG construction to ensure that human-centric challenges are not overlooked.

With this paper, we would like to advance the field of **explainable knowledge engineering** to allow KG stakeholders to rely appropriately on AI algorithms and use KGs with confidence [25]. This paper explores transparency issues from multiple dimensions, examining both the technical perspective, assessing the current state of explainable knowledge engineering models and techniques, and the user-centric perspective, focusing on the accessibility and acceptance of explanations. Our investigation of explainability encompasses both the inherent ability of models to elucidate their internal mechanisms (i.e., interpretability) and the techniques used to generate explanations across various models. Moreover, we examine their potential applications and integration in practical scenarios. To achieve this, we need to first gain a better understanding of emerging KG construction practices in the era of ML-as-a-service and develop human-in-the-loop approaches to ensure transparency and accountability throughout the KG lifecycle. This applies both to proprietary KGs used within organizations [8] and publicly available KGs like Wikidata [26], DBPedia [27], YAGO [28], and ConceptNet [29], which are extensively used by researchers and practitioners. As AI laws and regulations enter into force, the trustworthy credentials of such KGs will have to be systematically assessed and documented.

Our paper follows recent work that explores emergent neuro-symbolic AI architectures from a system-design perspective. Van Bekkum et al. [30] propose a taxonomy of hybrid (i.e., learning and reasoning) systems and discuss common architecture patterns and use cases. Building on their insights, Breit et al. [31] carried out a comprehensive literature review to add details to those patterns in terms of inputs, outputs, processing units, types of ML models and their training, types of knowledge representation and reasoning, but also transparency and auditability. One of their main findings is that most system designers do not consider these latter aspects at all, or, when they do, they do not evaluate them sufficiently. A third paper by Tamašauskaitė and Groth [18] draw from a survey of system papers to define a canonical KG construction process. Our work continues where they left off: starting from their KG construction process, we follow one of their main recommendations to map models and techniques for each step to provide additional guidance to researchers and developers.

Thus, we put forth the following research questions:

- **RQ 1:** What is the state-of-the-art/status of explainable automated KG construction?
- **RQ 2:** How do knowledge engineers and knowledge graph researchers understand their models and techniques and explain their output to stakeholders?
- **RQ 3:** Do the existing explainable models and techniques meet the requirements of knowledge engineers in practical use cases?
- **RQ 4:** What are the requirements of knowledge engineers and knowledge graph researchers for explainable approaches?

We analyze the KG lifecycle to identify tasks that are commonly automated with AI and those that still require human input and oversight and could potentially benefit from AI assistance. This work builds upon our previous study [23], in which we surveyed the state-of-the-art in explainable AI (XAI) to inform the design of XAI approaches that are practically useful for KG stakeholders such as knowledge engineers, subject domain experts, and users.

Furthermore, to extend our methodologies, we conducted an interview study involving 13 knowledge engineers and researchers from the knowledge engineering community. The interviews further explore topics such as their degree of understanding of models and techniques, their degree of automation, their transparency and explainability requirements, and various usage scenarios. Our main findings are:

1. There are tasks in KG construction, for instance, knowledge acquisition, where automation[1] is routinely used with promising results. At the same time, there are opportunities to use AI to assist other tasks, including ontology reuse, ontology evolution, ontology evaluation, and documentation, where (the latest) AI capabilities have remained under-explored.
2. While tasks around knowledge acquisition, taxonomy building, and data ingestion are often automated, human oversight is still needed to improve performance, establish trust, or comply with the law. In our review, we found little evidence of the integration of AI capabilities besides basic automation, no matter their level of interpretability, into standard knowledge-engineering tools and practices. Furthermore, our understanding of human-in-the-loop KG construction remains limited, with implications for user experience.
3. Comprehensive evaluations of XAI methods are lacking, with most studies focusing on simple ML models in lab settings, with mixed results [32–34]. The KG community, just like elsewhere in AI, needs to gain a better understanding of how people react and use explanations to build trust and boost technology adoption.
4. Knowledge engineers have varying levels of understanding regarding the models and techniques they use, with many expressing concerns over the opaqueness of black-box models. Data provenance and lineage tracking are recognized as critical, yet there are still gaps in the comprehensiveness and standardization of these practices. Evaluation heavily relies on human effort, highlighting the need for more robust and scalable methods. Additionally, effective communication of tool functionality and results to diverse stakeholders remains a significant challenge, requiring tailored approaches to bridge knowledge gaps and align expectations.
5. Current XAI solutions often fail to meet practical requirements, as their explanations tend to be insufficiently informative, overly complex, and lacking in stability and coverage. Furthermore, findings from the interview study highlight the need for explanations that are both clear and confidence-indicating, with a strong preference for natural language representations.

The remainder of this paper is structured as follows: Section 2 provides the background and related work, including an introduction to the KG lifecycle. Section 3 outlines the research methodologies, presenting the two-dimensional XAI taxonomy and use cases for literature analysis, as well as the foundation of the interview study. Section 4 explores the key findings from both the literature review and the interview study, with Sections 4.1 through 4.4 addressing research questions 1 to 4, respectively. In Section 5, we propose a blueprint for the design of explainable knowledge engineering models. Finally, Section 6 concludes the paper. To facilitate further research, we maintain a public repository[2].

2. Background

2.1. Transparency and Explainability of ML Methods

Transparency as an AI design principle stands for the need to clearly document and explain how an AI system makes decisions, how the data is collected, used, and governed, and how the system is evaluated and audited [35–37]. Achieving transparency in machine learning (ML) models can be accomplished through explainability. Although some ML models, like decision trees, are naturally interpretable, larger models, such as language models, are too complex to comprehend in the same way. To address this issue, researchers and practitioners have proposed many XAI frameworks, guidance, standards [38], techniques [39, 40], and evaluation metrics [41] for various models within the context of trustworthy AI. Typically, surveys on XAI models and techniques focus on aspects like problem

[1] In this paper we use AI assistance and automation interchangeably. While we acknowledge that not all automation in KG construction is AI, we argue that the use of AI brings about specific challenges with respect to transparency, accountability etc.

[2] https://github.com/bohuizhang/XKGC

formulation, taxonomies and classification, evaluation metrics, challenges, and future directions [38, 42–45]. For works that are more related to ours, Danilevsky et al. [46] conducted a survey on the state-of-the-art XAI models in natural language processing, which includes tasks that overlap with our work, such as named entity recognition and relation extraction. In the area of XAI and KGs, researchers have suggested using KGs to provide explanations. Tiddi and Schlobach's systematic literature review [6] focused on the integration of KGs into explainable machine learning, where KGs are used as domain knowledge for explanations. In addition to the technical perspective, Miller's review [47] provided a thorough examination of explainable AI through a sociotechnical lens, drawing from a variety of fields such as philosophy, cognitive science, and social psychology. Although previous studies have focused on some KG construction tasks and applications, a thorough review of the transparency and explainability of knowledge graph construction is still missing.

2.2. User Studies on Explainable AI

A deep understanding of the end-user requirements is essential in order to design trustworthy explanations, as explainability is a human-centric property [48, 49]. Preece et al. [50] give an analysis of stakeholders in XAI by examining the concerns of various stakeholders communities and digging into their different intents and requirements. Ras et al. distinguished different users of deep learning models into two groups and discussed their concerns: the expert users, who are engineers and developers building and maintaining the systems, and lay users, who are the end users and stakeholders [51]. Liao et al. [52] conducted interviews with UX and design practitioners working on various AI products through question-driven explanations. It is noteworthy that there is a lack of user studies on XAI involving knowledge engineers and knowledge graph stakeholders as end-users. Therefore, there is no consensus among design disciplines for XAI in relevant domains. Similar to our intents, Dhanorkar et al. [53] conducted an interview study on XAI towards AI researchers and stakeholders in industrial AI projects focusing on the AI lifecycle. Rong et al. [48] surveyed user studies through characteristics including trust, fairness, understanding, usability, and human-AI collaboration performance, and provided guidelines for both XAI researchers and practitioners on designing and conducting user studies. Similar to our interview study, Kim et al. [54] conducted an interactive feedback session in their interview study with the objective of understanding how explainability can support human-AI interaction. They mock up explanations that could be potentially used for AI application outputs in the field of computer vision to assess the participant's perception of existing XAI approaches and how participants use explanations during their collaboration with the AI. Automated and transferable evaluation, benchmarking, and comparison of XAI approaches pose open challenges, as explainability is often seen as a subjective property, necessitating auditing from multiple aspects [55]. On the other hand, human-centered XAI evaluations that take an HCI perspective remain critical in XAI evaluation, where rigorous evaluation procedures need to be established [56].

2.3. Human-Centric Knowledge Engineering

Knowledge engineering, the branch of AI concerned with building and managing knowledge-based systems [57, 58], has changed dramatically with the latest innovations in machine learning, natural language processing, and computer vision. The process of constructing a knowledge graph can take on various forms, but it usually involves acquiring knowledge, processing it, and deploying the knowledge graph [1, 18, 59]. And yet, as the most recent advances in natural language processing (especially LLMs) and generative AI demonstrate, the question of how to capture and encode domain knowledge into a computational representation remains as challenging as ever [60]. The technologies and end-user tools to support core knowledge-engineering tasks such as knowledge acquisition have advanced significantly to meet the scale requirements of modern KGs and to leverage the generative ability of sequence-to-sequence frameworks [61, 62]. AI copilots, which leverage LLMs, have also become involved in the KG lifecycle through conversational interactions [63], assisting knowledge engineers and users in a wide range of tasks. At the same time, the most effective approaches to knowledge representation still require human oversight at various levels [64, 65], but increasingly human input is in the form of enhancing or validating algorithmic suggestions [18]. The tasks of knowledge engineering require human-in-the-loop to a different extent and are considered human-centric [24, 66, 67]. These developments have resulted in improved methods and techniques to support the knowledge engineering process, with a growing group of participants and stakeholders, including knowledge

engineers and domain experts [64]. Witschel et al. identified human-in-the-loop patterns in hybrid learning and knowledge engineering activities, encapsulating them in two boxologies, where human agents function either as feedback-providers or feedback-consumers [67]. Back to 2002, after Holsapple and Joshi introduced the first collaborative approach to ontology design [68], various collaborative ontology engineering methodologies have been proposed, including tasks like ontology design and construction [69–72], ontology evolution [69, 72–74], and ontology evaluation [75, 76]. The tasks of ontology engineering continue to rely heavily on manual labor, and many of the reviewed works are outdated and pre-date the era of deep learning. There are evident challenges in improving the methodologies used in this process and adapting them to meet the requirements of automation, scalability, and transparency.

2.4. The KG Lifecycle

Building on the process from [18], Figure 1 shows that the KG lifecycle today consists of four stages with a mix of automated and manual capabilities and contributions from several stakeholder groups: knowledge engineering and machine learning specialists, subject domain experts, online volunteers, and crowdsourcing services, as well as developers of applications using KGs.

As the figure illustrates, KGs interact with AI capabilities in complex ways, involving multiple groups of people collaborating both with each other and with machines. Human-in-the-loop tasks in KG lifecycle increasingly use ML models with varying levels of interpretability. On the left side of the figure, at stage A, which is an entry point and essential step of the KG lifecycle, knowledge engineers and KG stakeholders (e.g., domain experts) will first determine the scope of work and the success criteria [77]. After that, at the second stage, knowledge graph construction, knowledge engineers and other specialists (potentially) reuse standard ontologies and build knowledge graphs from scratch through data lifting and knowledge extraction. Multiple data sources, structured and unstructured, are lifted into KGs using ML for named entity recognition [78], relation extraction [79], entity reconciliation [80], and many others. The ontology organizing the KG can be provided upfront or derived from the data itself, depending on whether there is a clear domain or available structured data with predefined types of entities and relations [18]. In this context, [20] discusses the need for more transparency with respect to data provenance and currency; both can affect whether application developers and end-users will be able to use the KG with confidence as a source of reliable, complete, unbiased, and up-to-date information. KGs can also be created on a larger scale through human collaboration, utilizing crowdsourcing platforms, collaborative-editing platforms, etc [1]. Crowd workers and volunteer editors have important roles in the KG lifecycle, especially in knowledge graph creation and updates, where annotation tasks such as quizzes and voting are often designed for leveraging their background knowledge [81–83]. While KGs constructed using these approaches may exhibit quality issues such as errors [84, 85], disagreement [86], bias [1], etc., crowdsourcing for supervised ML may have similar transparency challenges as the algorithms it complements. This is because the digital services commonly used for this purpose, e.g., Prolific and Mechanical Turk, are black-box, proprietary platforms with limited means to replicate or reproduce results [87]. Educating crowd workers in the process of performing crowdsourcing tasks is also a nontrivial task [82]. Interleaving explanations during this process could aid in educating crowd workers, enhancing their comprehension of the task, and ultimately improving output quality.

The result of knowledge acquisition is shown in the middle of the figure, where KGs are often linked to third-party data, reuse standard ontologies and identifiers, and are encoded as RDF, JSON, or other formats. On the right-hand side of the figure, KG maintenance (stage C) is prompted by source updates from stage B, and requirements, audits, and assessments from stage D. To further increase their completeness, correctness, and utility, KGs are refined by completion tasks such as link prediction and error detection and correction tasks, etc [88–90]. At stage D, there are a selection of use cases for KGs alongside other forms of AI. KGs are used as knowledge bases to query and reason upon, for instance in search [91], question answering [92, 93], and retrieval-augmented generation [4, 94]. Information can be obtained from a graph through deductive (e.g., logical rules) and inductive methods (e.g., as continuous graph embeddings) [1]. Both methods need to be transparent and accountable to the user [95, 96] to be trustworthy and compliant with laws.

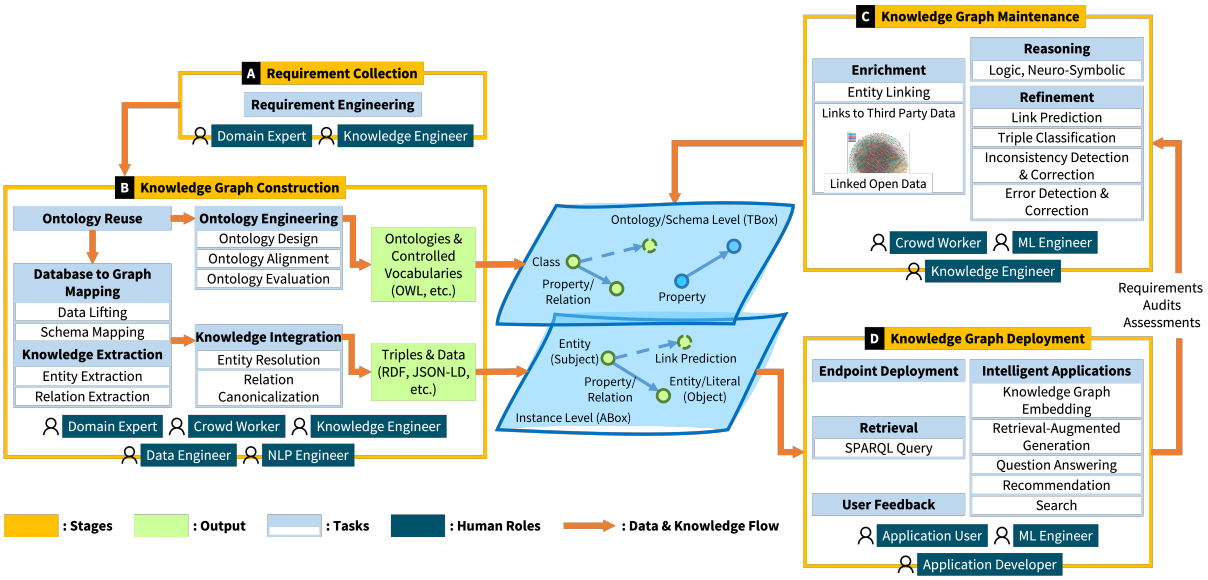

Fig. 1. The knowledge graph lifecycle today.

Knowledge Graph Construction	Transparency AI
knowledge graph construct*, knowledge graph develop*, knowledge graph complet*, knowledge graph refine*, knowledge graph reasoning, knowledge graph inference, knowledge engineering, named entity recognition, extract entit*, relation extract*, entity linking, entity matching, entity resolution, entity alignment, link prediction	transparent, transparency, interpretable, interpretability, explainable, explainability

Table 1

Keywords for the literature search query. Keywords from two groups were combined for query construction. '*' represents wild characters that can match any word suffix in the search.

3. Methodology

To address our four research questions, we employed a mixed methodology of systematic review and interview study. The systematic review involved collecting and analyzing literature on explainable AI in the context of knowledge engineering to gain insight into its current development. The interview study allowed us to directly explore the role of explainable AI in broader contexts, understand the needs of knowledge engineering and KG stakeholders for explanations, identify potential gaps and challenges in this field, and provide valuable insights for further research.

3.1. Literature Review

3.1.1. The PRISMA-guided Review

Following the discussion of the lifecycle, we carried out a PRISMA [97] literature review on databases including ACM Digital Library, IEEExplore, ScienceDirect, arXiv, SpringerLink, and Google Scholar. We searched for queries combining, on the one side, keywords related to trustworthy (mainly transparent and explainable/interpretable) and, on the other side, keywords related to KG construction tasks, as shown in Table 1. The search initially encompasses all keywords related to KG construction tasks, as depicted in Figure 1. We conducted a prototype search by examining the top 20 results generated by these keyword patterns. Subsequently, we eliminate keywords associated with tasks that do not yield hits within the top 20 results, thereby streamlining the review process. The search took place from October to December 2022 and resulted in more than 735K hits. We then took the

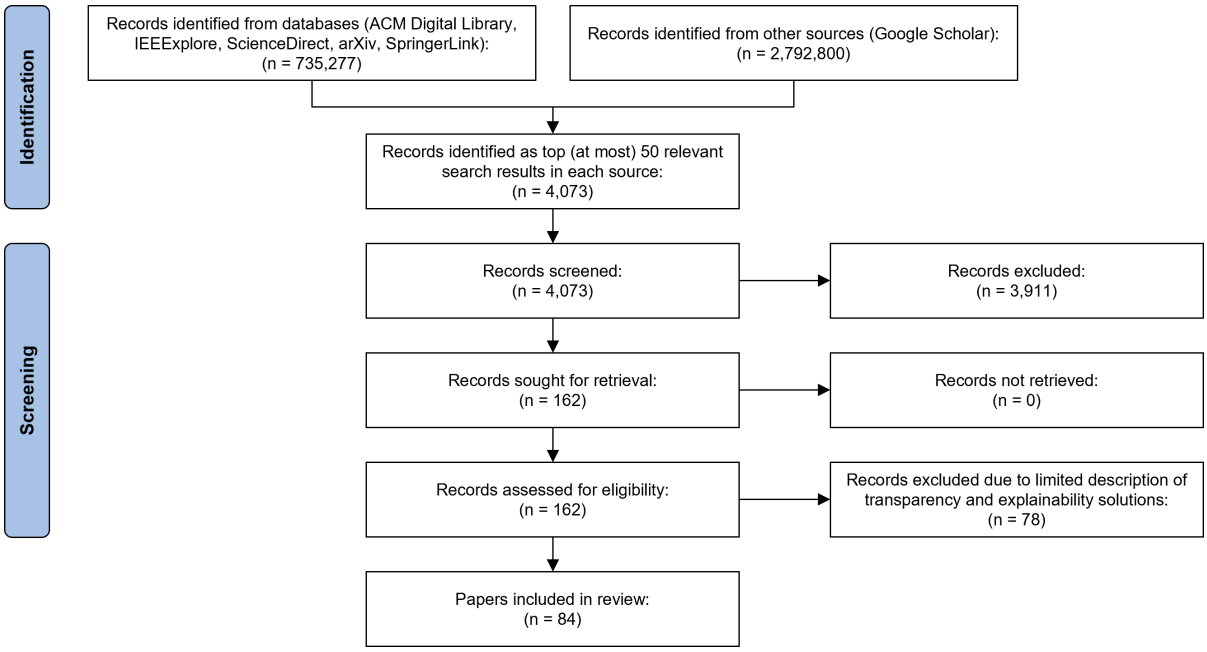

Fig. 2. The PRISMA flow diagram for systematic review.

Use Case	Intentions
Model selection and building	Help users understand the characteristics of ML models thus select the fitted model and build the pipeline
Model debugging	Detect errors may happen in the process and help model users avoid or fix the error
Understanding performance and contributing factors	Give explanations to the predictions and analysis to the contributing factors of the final results
Managing updates	Help users understand how the pipeline will change as data updates and help improve the results

Table 2

Summary of use cases of XAI methods in knowledge graph construction process and their related objectives.

top 50 hits per query, which led to around four thousand papers with duplicates[3]. The workflow of paper selection is shown in Fig. 2. We assessed relevance based on titles, abstracts, and keywords first, and in a second step, reviewed the text of the paper to select only those papers that proposed a solution to transparent and explainable KG construction, either as a whole process or for individual tasks. We discarded papers that only mentioned transparency and related concepts rather than putting forward a solution. The final corpus consisted of 84 papers. The papers were all published in the past ten years, which was to be expected given the term "knowledge graph" was coined in 2012 and is in line with other recent knowledge-graph surveys [18, 61].

3.1.2. Use Case Analysis

In addition to reviewing the existing work categorized in Section 4.1.1, we adopt use cases as an orthogonal dimension for literature analysis inspired by [98], recognizing that explanations serve diverse end users with varying needs across different scenarios and stages of the KG lifecycle. To derive XAI use cases in the KG lifecycle, we first examine use cases in the broader AI lifecycle and specific domain applications [98, 99]. We then map these use cases to practical scenarios and task spaces within the KG lifecycle, as illustrated in Figure 1. Specifically, we identify and present four key use cases, along with their objectives, in Table 2.

[3]The six platforms where we performed the search supported different query affordances. This means that in some cases, it was possible to build complex queries with multiple keyword options, whereas in others, we had to use separate queries to achieve the same results. We took the first 50 hits for each search query.

Use Case 1: ML Model Selection and Building When ML is incorporated into knowledge engineering, ML and knowledge engineers must select the proper models and build them. To help users evaluate and select suitable ML models, explanations should reveal the characteristics and limitations of the model, potential risks associated with its use, and its specialization for data or domains. In particular, they should address questions such as the model's capabilities, strengths and weaknesses, and data fitting. It is also important to determine if the model exhibits bias toward specific groups of data sources.

Use Case 2: ML Model Debugging One of the purposes of providing explanations for ML models is to facilitate debugging by allowing knowledge engineers to identify inaccuracies and flawed predictions and providing them with actionable information to correct them.

Use Case 3: Understanding Performance and Contributing Factors To ensure a thorough comprehension of performance, explainable knowledge graph construction pipelines should include the following elements:

- A clear understanding of the inference/reasoning process, which can be represented as rules, paths, etc.
- Identification and highlighting of the factors, important features, and supporting evidence that contribute to the final predictions.
- Provision of counterfactual interpretation through perturbation/permutation.

Use Case 4: Managing Updates Explainability is crucial for knowledge graph maintenance. When updates occur in data sources and contextual information, the knowledge graph can be updated by rerunning the construction pipeline, executing update or modification models, and so on.

To validate the use cases, we compared the use cases derived from the literature review to the ones collected from the interview study and found that the use cases derived from the literature review are mostly reflected through the interview study, and the latter also provide new ones, which we further discussed in Section 4.3.

After identifying use cases, we conducted further investigations into the capabilities of existing works with respect to these use cases. There are two main aspects to consider for this purpose. Firstly, we need to determine whether the reviewed methods have been applied in real-world scenarios of the given use case or could be adapted to suit them. Additionally, we need to consider whether the models have been trained and tested on real-world data. In the domain of knowledge graph construction, benchmarks and datasets are usually close to real-world KGs, such as Wikidata, DBpedia, and Freebase. The second aspect to consider is whether the explanations provided are understandable and satisfactory to the intended audience for the given use case. This can be determined if the work has done comprehensive evaluations that include metrics and human evaluations. Thus, we will evaluate the capabilities of the existing methods based on the following criteria:

- $\times$: It is not clear if the method is applicable to the given use case.
- $\star$: The method has potential for the given use case.
- $\bigstar$: The method has been applied to the use case but has not yet been integrated into toolkits or applications in real-world scenarios. Additionally, the explanations provided by the method have not been evaluated through user studies or evaluations.
- $\bigstar\bigstar$: The method has been integrated into toolkits in real-world scenarios. Furthermore, the explanations provided by the method have been tested through real-world studies with the target audience.

3.2. Interview Study

Besides the literature review, we conducted semi-structured interviews with the objective to (1) acquire a basic understanding of the current status of knowledge engineering models and techniques, including transparency issues and obstacles, (2) figure out gaps between existing solutions and practical knowledge engineering scenarios, (3) collect practical requirements for explainable capabilities, and (4) capture insights to design automated explainable knowledge engineering pipelines. Table 3 lists all participants and their background information[4]. In total, we interviewed 13 researchers and knowledge engineers from August to November 2023. All participants were recruited via

[4]As a knowledge manager, participant *K* is responsible for designing and consulting on taxonomies and ontologies, as well as communicating and educating about knowledge graphs through presentations, webinars, and writing.

contact lists of research events, a hackathon, and mailing lists hosted by W3C[5]. We maintained a balanced gender distribution among participants (6 females and 7 males) and ensured diverse coverage in experience, domain, and tasks. In terms of sector representation, 7 participants are affiliated with universities, indicating a relatively stronger academic background, while 2 are from research institutes and 4 from companies. The latter 6 participants are considered to have a stronger industry background with a focus on industry-related scenarios. Each interview lasted 35 to 50 minutes via an online video call, which involved the authors and the participants. The ethical clearance was granted from the Research Ethics Office of King's College London with ethics registration confirmation reference number MRSP-22/23-34456.

ID	Job Role	Experience	Domain	Tasks
A	Researcher	2	Culture	Ontology engineering
B	Researcher	9	Legal, finance, culture	Refinement, knowledge extraction, knowledge integration
C	Researcher	1.5	Culture	Ontology engineering
D	Knowledge engineer	23	Medicine, scholarly, industry, finance, etc.	Ontology engineering, intelligent applications
E	Researcher	3	Social science	Knowledge extraction, knowledge integration
F	Researcher	11	Industry, environment, tourism	Ontology engineering
G	Researcher	9	Public knowledge graphs	Knowledge extraction, refinement, knowledge integration
H	Researcher	17	IoT, medicine, insurance, tourism, etc.	Ontology engineering, datab transformation
I	Knowledge engineer	10	Customer data, public knowledge graphs, geography	Knowledge extraction, enrichment, data transformation
J	Researcher	10	Scholarly, cross-domain	Ontology engineering, knowledge extraction
K	Knowledge manager	3	Cross-domain	Ontology engineering
L	Researcher	20	Mobility, manufacturing	Ontology engineering, knowledge extraction, data transformation
M	Researcher	11	Biology, property, medicine, legal, energy, history	Knowledge extraction, refinement

Table 3

Background information about interview study participants includes the job role, experience working with KGs and knowledge engineering in years, domains of KGs, and tasks involved in the KG lifecycle.

3.2.1. Interview Questions

Table 4 presents all the interview questions organized by topics and the order in which they were asked. The questions addressed various topics, including the understanding level of models and techniques, degree of automation, data provenance and lineage, trust, evaluation and human intervention, explainability, and associated risks. The design of interview questions incorporated multiple factors, drawing from previous interview studies on explainable AI in other fields [53, 54], taxonomies and surveys of transparency and explainability [48], and the Explanation Ontology [100] to ensure comprehensiveness. We adapted these trustworthy factors to the context of knowledge graph construction. Firstly, we asked questions about the research background, including experience and domain, to acquire demographic information. Next, we asked about the participants' experience and understanding of the models and techniques they use. This foundation allowed us to assess the extent to which transparency is an issue and its impact on their practical work. Given the importance of data provenance as a dimension of transparency [101], we include questions specifically about this. To examine the human role in knowledge engineering and gain insight into human factors, we asked questions related to the evaluation of results and how humans interact with the pipeline, providing oversight and intervention. Inspired by [53], we designed questions about explanation scenarios and use cases. These questions delved into scenarios where participants explain results or models to their stakeholders, seeking to identify explainability concerns, challenges, and requirements. Finally, we addressed risk concerns that might arise if transparency and explainability are provided with current models and techniques, ensuring a comprehensive understanding of potential issues.

XAI Example Discussion Furthermore, by selecting examples from the previous literature review, we designed XAI examples and facilitated discussions on their usefulness, faithfulness, and acceptance. This approach directly connects stakeholders in the context of knowledge engineering with existing literature methods, highlighting the pros and cons of current explainable solutions and the gaps between these solutions and practical needs, given the limited application of existing XAI approaches in real-world knowledge engineering scenarios. The XAI examples

[5] semantic-web@w3.org, public-lod@w3.org

were directly selected from the reviewed papers. We first identified papers that provided examples of explanations, such as visualizations of attention weights, graph paths, and tables of reasoning rules. We then randomly selected two papers per task as examples for participants to discuss. During XAI example discussions, participants were first asked to select one (or two, if time permitted) task that they were familiar with. We then provided two examples of two explainable approaches to the selected task. Each example was presented on a slide, consisting of the input, output, and explanations as provided by the original publication. Table 8 lists the examples we selected, along with their representations and citations. After reviewing the examples, participants discussed the usefulness and acceptance of the explanations, such as whether they found the explanations helpful and whether they would accept them in their work scenarios or expose them to stakeholders, such as domain experts and users. Moreover, they were encouraged to identify defects in the explanations and suggest improvements or alternative solutions to make the explanations more acceptable. During this process, participants were free to ask questions about the provided examples, and we responded based on the original publication.

Topics	Questions
Background Information	What is your job title? How long have you been working on knowledge engineering and knowledge graphs?
Domain & Tasks	Can you give a brief description of your work with knowledge graph construction, including: • an introduction to the knowledge graphs, their types and domains, • tasks in which you have been engaged, such as knowledge extraction or completion?
Status	Could you please briefly describe the models and techniques that you use for the tasks you mentioned? • Are they fully automated or incorporate human efforts, e.g., human-in-the-loop? • Are they explainable or transparent? And why? • How do you perform the model selection?
Understanding	Do you understand, or do you need to understand how the automated components work in detail? • What are the obstacles to understand the performance of the component or the results it generated? • If not understood, will the opaqueness of the toolkits impact your work?
Data Provenance & Lineage	• Do you know where the data comes from? • Do you keep track of all operations that have been carried out? • How do you keep track of data provenance and lineage?
Evaluation & Human Intervention	• How do you verify or evaluate the results generated by the automated components or the pipeline? • Are there any mechanisms to help you? • If you could verify the results, is there any way that you can correct or modify them? • What kind of intervention do you take? Explain when and how you perform the oversight.
Explanation	Do you explain to another person how the automated components work or the generated results? • To whom do you explain the components or results? • What type of content do you explain? • How do you explain the results? Do you adopt any methods to help you deliver the explanation? • Do you encounter any challenges in this process?
Use Case	In what scenarios would you need the pipeline to give you an explanation?
XAI Example Discussion	Please select one of the following tasks, we will provide explainable examples and we can discuss them: (1) entity extraction, (2) relation extraction, (3) entity linking, (4) link prediction, (5) inconsistency detection
Requirements	After answering and thinking about the above questions, how would you envision a solution? • What kind of information do you hope to be provided by explainable pipelines? • What is your preferred form of explanation? • How will the explanations help your work?
Risk	What is your concern about the risk of explainability or transparency?

Table 4

The list of interview questions. The XAI example discussions are accompanied by slides introducing and showing explanations, and the following questions in this part are mostly intrigued by the responses of participants.

3.2.2. Coding and Analysis

The interviews were recorded using Microsoft Teams and transcribed with its automatic transcription services. The transcripts were then further cleaned and edited by the authors to remove repeated words, pauses, filler words, and to recover errors such as software names and abbreviations. The edited transcripts were coded into keywords and patterns, consisting of phrases and sentences. We employed three levels of coding strategies for different types

of questions. First, for questions related to background information, domain and tasks, and status, we used in vivo coding, extracting the exact words from the transcripts. For questions on data provenance and lineage, evaluation and human intervention, explanation scenarios, and requirements, we extracted the phrases and identified patterns such as operations, methods, and examples. Finally, for questions on understanding, XAI example discussions, and risks, we extracted patterns such as comments and suggestions, and coded the attitudes and beliefs towards the explainable examples. To analyze the coded data, we grouped identical and similar content into clusters of thoughts and insights, and counted the occurrence of each cluster. We also highlighted quotes to provide important supporting evidence, insights, and original ideas.

4. Findings

4.1. The Status of Explainable Automated Knowledge Engineering

4.1.1. The State-of-the-Art Explainable Models

We classified the papers reviewed with respect to the KG construction tasks they addressed and their approach to explainability, starting with categories widely used in the literature. For explainability, we started with what is explained: *local* (data point) vs. *global* (outcome); and when: *post-hoc* (after prediction) vs. *self-explaining* (while predicting). We then added another layer for post-hoc methods, splitting the methods into two subgroups: *model-specific* (specific to one or a group of models) and *model-agnostic* (can be applied to any model).

The results are presented in Figure 3 and visualized using a Sankey diagram in Figure 4. At a glance, the papers do not cover the entire KG lifecycle. Most papers are concerned with knowledge acquisition via entity extraction (as a source of classes and instances in KGs) and relation extraction (as a source of property classes, but more importantly connecting entities to each other through properties), or with curation and maintenance via entity resolution (consolidating the data that refers to the same entities) and link prediction (suggesting missing or emerging facts). Besides the four core tasks in the bottom half of the figure, we found one paper dealing with the evolution of the KG schema or ontology [180] and another one about detecting and explaining inconsistency in KGs [181]. We note that link prediction was by far the most popular task, and that a majority of papers dealt with curation and maintenance rather than building a KG for a particular purpose. This is somewhat concerning, as many applications of KGs are in enterprise contexts [8], where the first step is to build a computational representation of the enterprise's data, which is stored across various systems and modalities. We argue that for the tasks not included in the review, there are several potential reasons why almost no papers were found. Many of these tasks still rely heavily on manual work and human oversight and have not yet been automated, as we will later verify based on interview results. This includes tasks such as ontology reuse and ontology design. Additionally, there are tasks where automation, such as the use of LLMs, has been employed, like ontology alignment [185] and data lifting from databases, but explanations have not been considered.

A second high-level observation is the balanced split in the chosen format for explanations. Methods based on input and generated features use attention weights [124, 131], words [117, 118], attributes [102], etc. to generate explanations, which can be numerical, textual, or visual. By contrast, methods based on human-understandable background knowledge provide explanations in the format like logical rules [164], reasoning paths [160], and structured contextual information [120] as explanations. Given that we are interested in explanations that are accessible to knowledge engineers and subject domain experts, it would be interesting to evaluate if their familiarity with knowledge representation and/or the subject domain impacts how useful knowledge-based explanations are compared to feature-based ones, which sometimes require an understanding of machine learning. At the same time, explanations are generated in a different way for each of the four core KG construction tasks in the bottom half of the figure.

Entity Extraction For entity extraction, explanations often leverage contextual cues such as triggers [117, 118] and patterns of words [167], utilizing attention mechanism [186] and saliency map techniques. One notable work is myDIG [187], a human-in-the-loop system that compiles sophisticated rules written by domain experts into SpaCy rules for backend execution. This reduces the barrier for domain experts to interact with the machine and minimizes training effort. Additionally, myDIG records extraction provenance, allowing users to explore the downstream effects of their specifications. Another type of explanation used for entity extraction is example-based explanations,

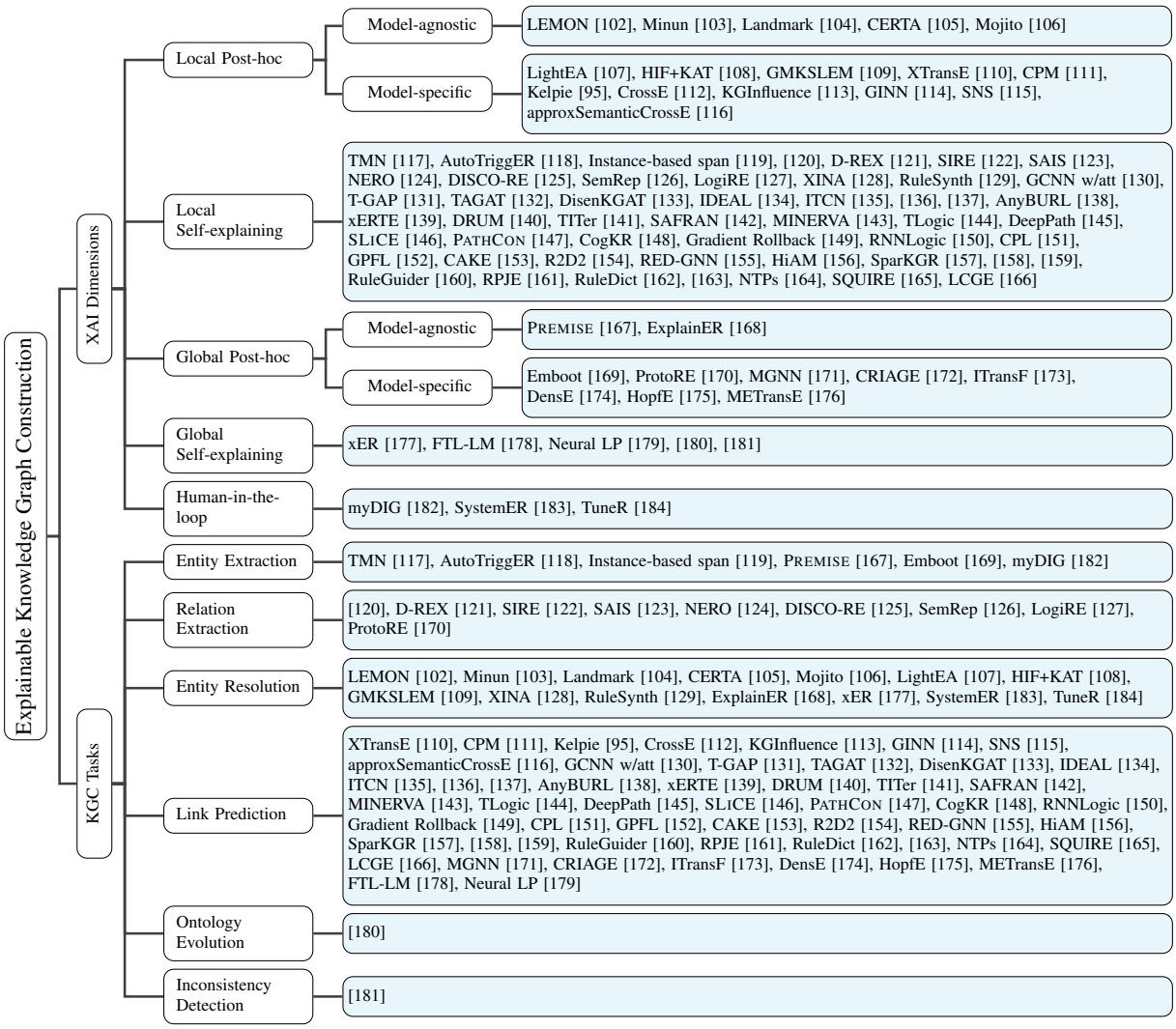

Fig. 3. Taxonomy of Explainable Knowledge Graph Construction. A summary table of the reviewed papers is also available in our previous work [23] and the GitHub repository.

which rely on training instances [188]. In Ouchi et al., similarities between pairs of candidate(s) and the training instances are computed, with the term having the highest derived label probability being returned [119].

Relation Extraction For relation extraction, explanations frequently employ contextual information from the input, such as words and sentences, similar to entity extraction. The attention mechanism is a prominent principle among relation extraction methods, with 4 out of 9 studies using attention weights and their associated input context to generate explanations. For instance, NERO uses word-level attention to calculate matching scores between sentences and generated rule patterns, where attention weights represent word importance for constructing attention-pooled rule/sentence representations [124]. SIRE [122] employs the attention mechanism in both the evidence selector [189] to identify supporting evidence and in the logical reasoning module [186]. In light of research questioning the validity of attention as faithful explanations, Shahbazi et al. adopted a mixed explanation mechanism extended by saliency, etc [120]. Another prevalent type of explanation in relation extraction involves relation learning/logic rules. Beyond NERO, LogiRE integrates a rule generator and a relation extractor, optimizing these modules using the expectation-maximization algorithm for document-level relation extraction [127]. Diverging from text-based explanations, ProtoRE learns prototypes for each relation from contextual information, exploring the intrinsic semantics

Fig. 4. Sankey diagram illustrating the categorization of methods. The left column represents KGC tasks and the right column represents XAI taxonomy. The total count for each category is indicated next to its label.

of relations and visualizing them as geometric explanations [170]. Under optimal conditions, prototypes are unit vectors uniformly dispersed on the surface of a unit ball, with datasets clustered around each prototype vector.

Entity Resolution There are two primary types of explanations for entity resolution: entity matching (EM) rules [108, 129, 183, 184] and (ranked) attributes of the entity pair with relevant scores [102, 104–106, 168]. EM rules, represented in forms such as disjunctive normal form and general boolean formula, are commonly used in EM systems to enhance interpretability [190]. For automatic EM rule-based models, Yao et al. proposed a framework consisting of Heterogeneous Information Fusion for learning feature representation from unlabeled data and Key Attribute Tree for interpretable EM decision making [108]. This framework translates decision trees into EM rules, making explanations more accessible to domain experts. RULESYNTH, proposed by Singh et al., formulates the rule discovery problem in entity matching as a program synthesis problem [129]. They adopted a more concise and interpretable form of General Boolean Formula to represent EM rules and proposed a novel rule synthesis algorithm. In contrast to EM rules, using attributes and their relevant scores as explanations focuses on uncovering the contribution and importance of each attribute or combinations of attribute sets in the decision-making process of entity matching. Most works employing this representation adopt perturbation-based methods [191]. By applying LIME perturbation to the entity resolution problem [39], these methods introduce perturbations, such as removing words and reducing similarity between entity pairs, to analyze variations and compute predefined importance scores. Conversely, they also explore the insertion of input attributes into entity pairs to assess whether such modifications can increase similarity between non-matching pairs.

Link Prediction Most explainable link prediction methods leverage the topology and reasoning capabilities of knowledge graphs (KG). Rule- and path-based methods have become the predominant forms of explanations, achieved through various approaches such as random walk-based methods [138, 144, 178], reinforcement learning agents [143, 145], and perturbation-based methods [95, 172]. A significant body of work utilizes reinforcement learning (RL) for reasoning over knowledge graphs and searching for paths to explain link prediction results [136, 141, 143, 145, 151, 154, 157, 160]. These models typically comprise knowledge graph environments and policy network agents. The knowledge graph environment transitions elements within the graphs (e.g., entities, relations, queries) into RL agent elements, where states are usually entities (in practical terms, embeddings) and queries (subject entities and relations); actions are typically outgoing edges/relations; transitions map current entities and their outgoing edges to their neighboring nodes; and rewards are heuristic indicators, awarding 1 when the agent reaches the correct target entities. Policy networks then maximize the expected reward to perform path finding. Variations exist in environment transitions, rewards, and the parameterization of the policy function. For example, R2D2 [154] and RuleGuider [160] employ multi-agent architectures. R2D2 uses two agents, with one arguing the fact is true and the other arguing it is false, feeding their arguments into a judge network. RuleGuider uses a relation

agent and an entity agent that interact to generate paths fed into a rule miner. Perturbation-based methods are also applied in link prediction, similar to those used in entity resolution. CRIAGE [172] introduces graph perturbation by removing a neighboring link from the target fact to assess the influence of the fact and by adding a new, fake fact to evaluate model robustness and sensitivity. Another prevalent method in explainable link prediction models is the attention mechanism, used in 16 out of 53 total link prediction works. For instance, XTransE employs attention values on items to reveal the relevance between different property-value pairs and the current prediction, which are then ranked to identify the most relevant triples [110]. In xERTE, Han et al. propose a temporal relational graph attention layer that calculates query-dependent attention scores for each edge [139]. These scores propagate to each node's prior neighbors, pruning the inference graph using edge contribution scores. The pruned graph, with node attention scores and edge contribution scores, is used to produce the explanations.

Human-in-the-loop There are very few papers considering human inputs or oversight, which are critical in trustworthy AI frameworks and guidance [192]. In the few cases of human-in-the-loop systems, human input often involves the provision or revision of rules for tasks such as entity extraction [182] and entity resolution [183, 184]. In myDIG [182], a GUI-based rule specification system is provided for domain experts to input expressive entity extraction rule sets without programming. SystemER [183], which adopts an active learning methodology, learns explainable entity resolution logical rules and offers functionalities for domain experts, both with and without programming backgrounds, to verify and customize the learned models in feature engineering to ensure extensibility. For generating entity resolution rules, TuneR [184] involves developers (i.e., coders, scientists, and domain experts) in tuning rule sets by defining the contribution of optimization metrics. The framework defines interpretability-related metrics as the preference between the number of rules in the rule set and their overlap. All three approaches use an ensemble of rules to achieve high precision. Several factors influence the success of these human-in-the-loop approaches, some of which have been considered in these three systems. One critical factor is balancing the minimization of training with the extent of human intervention. More human intervention can reduce training efforts, which require feeding more data thus extending training time. Conversely, increased training efforts can reduce human intervention, thereby minimizing unnecessary human labor and avoiding time-consuming and error-prone trial-and-error processes. Another factor is the degree of operational freedom given to users. The complexity of functions and the freedom of operations provided to users affect the time required to educate them. The design of functions should enable users to maximize their input to produce high-quality work while minimizing the time needed to familiarize themselves with the tool. Providing too few intervention options might hinder users from fully expressing the correct input, thereby increasing human effort. These factors are crucial when designing human-in-the-loop systems, and more user studies, especially for knowledge engineers and knowledge graph stakeholders, are needed to explore them further.

Evaluation of explanations We also collected and analyzed the evaluation of explanations. A primary observation is that most XAI approaches have not been thoroughly and/or comprehensively evaluated. The majority of methods (58 out of 84) do not perform any evaluation on explanations or only use anecdotal evidence by visualizing and commenting on a limited number of cases of explaining outcomes intuitively. There are efforts to design metrics to evaluate explanations. 17 works adopted metrics to evaluate their explanations, and most of them are task-dependent. Shahbazi et al., [120] created a ground-truth explanation set and computed the Kendall Tau correlations for the sentence importance scores for the annotated test set. approxSemanticCrossE [116] proposed explanation evaluation metrics targeting the link prediction tasks, which calculate the ratio of triples for which the model can generate explanations (recall) and the number of explanations, on average, for each prediction (average support). In gradient rollback [149], Lawrence et al. adopted the "RemOve And Retrain (ROAR)" [193] evaluation paradigm to evaluate the faithfulness of the explanations.

12 studies use human evaluation, detailed in Table 5. We identified 5 types of evaluation tasks commonly adopted in these studies. The most frequent tasks involve asking participants to compare model-generated explanations with those from baseline models and to judge the relevance and correctness of a set of examples. Various metrics are used in human evaluations. One approach is to have participants rate the usability, reliability, and trust of explanations in a survey. A notable example in this group is SQUIRE [165], which annotates BIMR-based interpretability scores [194] for paths generated by their models and baseline models. Another group of methods measures the accuracy or precision of user predictions with or without provided explanations. The backgrounds of human evaluators are

Evaluation Tasks	Methods	Number of Participants	Background
Comparing model-generated and human-provided explanations	AutoTriggER [118]	/	crowd-workers
	D-REX [121]	3	crowd-workers
Judging the relevance and correctness of explanations (examples)	AutoTriggER [118]	See above	
	Emboot [169]	2	domain experts
	xERTE [139]	53	*
	DRUM [140]	2	CS students
Comparing explanations generated by different models	D-REX [121]	See above	
	RuleSynth [129]	27	CS researchers
	RuleGuider [160]	/	crowd-workers
	DRUM [140]	See above	
	SQUIRE [165]	/	authors
Survey with questions measuring the quality (usability, reliability, trust, etc.) of explanations	Kelpie [95]	44	/
	SQUIRE [165]	See above	
Evaluating the accuracy or precision of user predictions with or without explanations	R2D2 [154]	44	/
	[137]	/	domain experts

Table 5

Works that use human evaluation to analyze explanations. '*' indicates that no group label is provided, but other detailed background information of participants is reported. '/' means 'not reported' in the paper.

varied, including domain experts, such as e-commerce experts in [137] and linguists in Emboot [169], people with technical backgrounds, and laypeople such as crowdsourcing.

From the above observations, we identified several issues with the evaluation methods. First, reporting a limited number of examples selected based on the researchers' intuition can be biased and not sufficient for robust verification [55, 195]. Since not all results have satisfied explanations generated, another issue is that the ratio of results for which the model can generate satisfied explanations is not commonly reported. In our interview study, we found it to be a crucial factor that might influence the user's trust in the XAI models.

4.1.2. Use Cases and Capabilities Measurement

Use Case	Capabilities				Methods
	Local Post-hoc	Local Self-explaining	Global Post-hoc	Global Self-explaining	
Model Selecting and Building	★	☆	★	×	ExplainER [168], CPM [111], Kelpie [95]
Model Debugging	☆	☆	★	×	LEMON [102], ExpalinER [168], D-REX [121], Instance-based [119], [182], TuneR [184], CRIAGE [172], Kelpie [95], SparKGR [157], GCNN w/att [130], MINERVA [143]
Understanding Performance and Contributing Factors	★★	★★	★★	★★	All papers
Managing Updates	☆	★	★	★	ExplainER [168], CPM [111], TuneR [184], Abstraction [181], TLogic [144], Emboot [169], SystemER [183], RNNLogic [150], [137] Neural LP [179], FTL-LM [178], ITCN [135], CPL [151], SQUIRE [165], MGNN [171], DRUM [140], ProtoRE [170], CRIAGE [172], TITer [141], PATHCON [147], METransE [176], RED-GNN [155], [180]

Table 6

Capabilities of XAI methods in knowledge graph construction. Symbols are referenced from Section 3.1.2: ×: applicability to the given use case is unclear; ☆: method shows potential for the given use case; ★: method has been applied to the use case but is not yet integrated into toolkits or real-world applications. Explanations provided by the method have not been evaluated through user studies or any other evaluation methods; ★★: method is integrated into toolkits in real-world scenarios, and its explanations have been tested through real-world studies with the target audience.

The capability of various explainable techniques for each use case is shown in Table 6. In general, the reviewed literature indicates that global post-hoc methods, especially model-agnostic ones, have the potential to address all use cases. Local post-hoc methods have also demonstrated similar potential across all use cases. Although no global self-explaining methods were identified for the first two use cases, this does not imply that these methods lack potential for model selection, construction, and debugging. Instead, they are suitable for providing model analysis due to their global assessment capabilities. Among the use cases, all except for understanding performance and contributing factors have received less attention and research. This could pose challenges when integrating developed methods into real-world applications, making it essential to address these gaps.

Use Case 1: ML Model Selection and Building Most model-agnostic methods, such as explainers designed for knowledge graph embedding models, and some model-specific methods, have the advantage of providing explanations across different models and facilitating comparison. While some of the reviewed works have demonstrated their applicability in this use case, most have not emphasized addressing concerns related to model selection and comparison. A notable example that covers this use case is ExplainER [168], which offers a mechanism for model analysis. The analysis engine of ExplainER comprises multiple explanation models and techniques (LIME [39], Anchors [196], BRL [197], and Skater [198]) that are independent of any entity resolution models. For link prediction, explainable methods such as CPM [111] and Kelpie [95] can be used with any embedding-based link prediction models, allowing for comparison across different embedding models. The main gap for current models in this use case is not solely related to model design and architecture, but also to better documentation. One potential solution is to document an interactive model card [199] that lists all the necessary information regarding explainability. For instance, for explainable link prediction models, this could include the ratio of faithful and correct explanations generated for each embedding model and a comparison of generated explanations for the same input.

Use Case 2: ML Model Debugging Some works provide analyses of errors. For example, the instance-based explainable method performed error analysis using relevant examples to identify factors causing model confusion [119]. ExplainER visualized representative explanations to highlight where the model fails [168]. D-REX conducted error analysis on explanations alongside model predictions, further revealing the model's error detection capabilities [121]. Pezeshkpour et al. demonstrated the potential application of CRIAGE for automated detection of erroneous triples in knowledge graphs. Their approach focused on identifying triples with the least influence on the model's prediction of the training data [172]. Similarly, Rossi et al. highlighted the ability of Kelpie to uncover bias and imbalance in data, enabling researchers to correct it. However, although these works provided analyses of errors, most did not offer actionable steps for rectifying the identified issues. This could be achieved by providing options to adjust parameters, model architectures, and leverage external sources such as human knowledge. Human-in-the-loop methods exemplify approaches for correcting errors and improving model output, such as manually correcting rules by domain experts in rule-based explainable systems [182, 184]. One approach following this line is to offer local actionable information, such as suggestions for correcting predictions directly. A future direction in designing local explainable methods would be to help users identify error cases and enable corrections at the data point level.

Use Case 3: Understanding Performance and Contributing Factors The majority of the reviewed work performed well across various tasks. As detailed in Section 4.1.1, a range of representations are employed to understand the inner workings of models and the factors contributing to their outputs. For knowledge extraction tasks, such as entity and relation extraction, models provided supporting evidence from the source data (e.g., text) to aid in predictions [117, 118]. Similarly, for knowledge integration tasks like entity linking, attributes of entities were selected through mechanisms such as matching or non-matching votes, as demonstrated in [102, 104]. Explainable link prediction models offered rules [129, 138, 160] and paths [143, 145, 151, 154] to illustrate the reasoning process, as well as subgraphs [148] to measure the influence of nodes and edges. Notably, rule-based methods are prevalent across all tasks due to their concise and straightforward representation and their ability to generalize to new data.

Use Case 4: Managing Updates Global explainable methods such as rule-based methods [144, 184] can potentially express the model evolution through modifications in their global explanations. Similarly, visualization-based explanations [170, 176], where users can compare different versions of visualizations, can also provide valuable insights when managing updates to knowledge graphs. Models that provide local explanations, such as inductive models [140, 141, 147] and perturbation-based models [172] could track differences for specific instances or groups

of instances. Very few of the models directly implemented this capability, but most of them could be potentially extended to support this use case. For rule-based explainable methods, a straightforward way to manage updates is to use the generalization ability of existing rules and perform inductive reasoning. For instance, TLogic [144] stated that the temporal rules they generate are applicable to any new dataset, as long as the new dataset covers common relations, even in cases where new entities appear. Zhang et al. [137] also emphasized the benefits of transferable rules. Their model could generate reusable rules to accelerate the deployment of a knowledge graph to new tasks or systems. In addition to directly transferring rules to new data, rules can also be updated. For example, RNNLogic [150] used an EM-based algorithm to update rules. Once the explanation rule sets were updated, to gain more insights, the users could compare two sets of rules and see what changes the new data had brought in. Similar strategies can be applied to other explanations, such as the visualization of attention weights and embeddings.

4.2. Explainable Knowledge Graph Construction in Practical Scenarios

We now report on the interviews. We first present the current status of knowledge engineering tools in practical scenarios, focusing on the degree of automation and the level of understanding that knowledge engineers have towards these tools, as well as aspects including data provenance and lineage, evaluation, and human intervention. By addressing a series of sub-questions, we aim to gain a basic understanding of these critical transparency factors from our interview study. This foundation will enable us to delve deeper into identifying the desired properties of explainable models and techniques.

Sub-questions for Interview Study	Summary of Findings
How much human effort is leveraged in the knowledge graph lifecycle?	Among the participants, the majority engage in manual (38.5%) and semi-automatic (38.5%) work, while a minority (23.1%) rely exclusively on automation for their tasks.
What is the level of understanding of the models and techniques?	46.2% of participants emphasized the impact of opaqueness on their work and the importance of understanding the models, while the remaining were less concerned with transparency. All participants exhibited a high level of understanding of their models and techniques.
Do knowledge engineers know where the data comes from?	92.3% of participants reported knowing their data sources, which varied by provider. Many received multi-sourced data depending on their projects.
How do people keep track of data provenance and lineage?	69.2% of participants actively tracked data provenance. Participants prefer standard representations using standardized languages and vocabularies.
How do knowledge engineers evaluate the results?	Most participants rely on human evaluation at varying levels to assess results.
What do people do when they find the results incorrect?	Human intervention is essential to addressing machine limitations in knowledge graph construction, occurring equally across three stages: pre-processing, in-processing, and post-processing.
How do people explain to others their models and results?	92.3% of participants have experience explaining models and results, primarily using example-based explanations. The main challenge when providing explanations is bridging gaps in background, knowledge, and requirements between knowledge engineers, researchers, and stakeholders.
What are practical use cases of XAI models?	The interviews revealed two novel use cases: enhancing human-machine interactions and uncovering previously unnoticed insights.
Do current explainable solutions meet the requirements for practical use cases?	The examples rarely met participants' requirements, with only 17.9% feedback responses being positive and nearly half negative. Participants cited concerns about informativeness, complexity, stability, and coverage of the explainable models.
What are characteristics of an explainable method that knowledge engineers and researchers expected?	Two key requirements were identified: a confidence indicator for models, explaining how confident the models are when producing results, and the use of natural language as the explanation format.

Table 7

Summary of key findings from the interview study.

4.2.1. Automation and Understanding

How much human effort is leveraged in the knowledge graph lifecycle? Among the participants, the majority engage in manual (38.5% of participants) and semi-automatic (38.5%) work, while a minority (23.1%) exclusively utilize automation for the tasks they work on. From the perspective of task execution in ontology engineering, participants predominantly employed manual and/or semi-automatic methodologies. These approaches necessitate extensive communication and collaboration among knowledge engineers, domain experts, and stakeholders, often facilitated through semi-structured interviews. Conversely, for tasks related to knowledge extraction and completion, participants demonstrated a preference for automated models and techniques. Methods, which focus on tasks like data transformation that lifts other formats of data into RDF triples through RML mappings[6] and tools like SPARQL Anything [200], always involved the manual creation of the mappings. One participant assessed the performance of leveraging language models in generating such mappings. Language models in knowledge engineering have enhanced automation due to their user-friendly nature, characterized by simple natural language input and output, which require fewer specialist skills. However, their opacity and tendency to generate hallucinations impact their trustworthiness. When evaluating the outcomes of models, such as the triples generated by knowledge extraction models, human evaluation is always necessary. This is particularly crucial when dealing with new domains and data, where datasets are lacking.

What is the level of understanding of the models and techniques? Participants had varying opinions regarding the impact of the opaqueness of models and techniques and the necessity to thoroughly understand them. 46.2% of them felt that opaqueness did impact their work and emphasized the importance of understanding the models. As participant *A* highlighted, this importance extends beyond merely explaining why the models produce certain outputs. It also involves helping humans "understand the extent to which these outputs can be trusted"[7] and determining "how they might need to change the way they interact with the model". Participant *B* also noted that the opaqueness of the models and techniques might complicate evaluations, as it becomes challenging to determine how specific inputs influence the outputs. In contrast, the remaining 53.8% of participants were less concerned with transparency issues, feeling that opaqueness was not a significant problem. They provided several reasons. Two participants stated that only the model's performance and the final quality of the output knowledge graphs mattered to them. Since they primarily deal with public datasets and transparency and explainability are not within their research scope, they pay less attention to these topics. Three participants mentioned that their tasks are predominantly manual, so transparency and explainability are less applicable. Some participants noted that even collaborative projects require some level of explanation for better communications and outcomes between human agents.

All 13 participants demonstrated a relatively high level of understanding towards the models and techniques they used, particularly when these models and techniques were open-sourced and/or came with documentation (e.g., publications, technical documents), or if the models were self-developed. Two participants mentioned that it is not always necessary to delve into the code level, and sometimes it is challenging to fully comprehend how the models make decisions. However, it is crucial to gain a conceptual understanding of the mechanisms of specific components, their technical limitations, and underlying assumptions. 53.8% of participants reported no significant obstacles in understanding the models and techniques. For the remaining 6, the challenges of understanding the models and techniques varied. The primary obstacle, mentioned by 3 participants, was the difficulty in understanding errors produced by models, their causes, and how the models arrived at decisions. Participant *B* pointed out that models could be difficult to understand due to their mathematical complexity and insufficient background knowledge in ML and NLP, particularly when it comes to understanding the inner workings of LLMs. Participant *E* noted the difficulty in determining the optimal size of data and model parameters to train models effectively or to transfer them to another domain or input type. Participant *L* emphasized the challenge of evaluating both the correctness and the completeness of results, noting that both aspects are critically important. Additionally, understanding "what level of quality is good enough for the task" is also challenging. To address these challenges, participant *A* suggested designing models to provide additional outputs that help in understanding the models. This is somewhat achieved by works in the literature review, such as models leveraging attention mechanisms that output attention weights [117],

[6] https://rml.io/specs/rml/

[7] We use double quotes to indicate that the quotes are the original words of the interviewees.

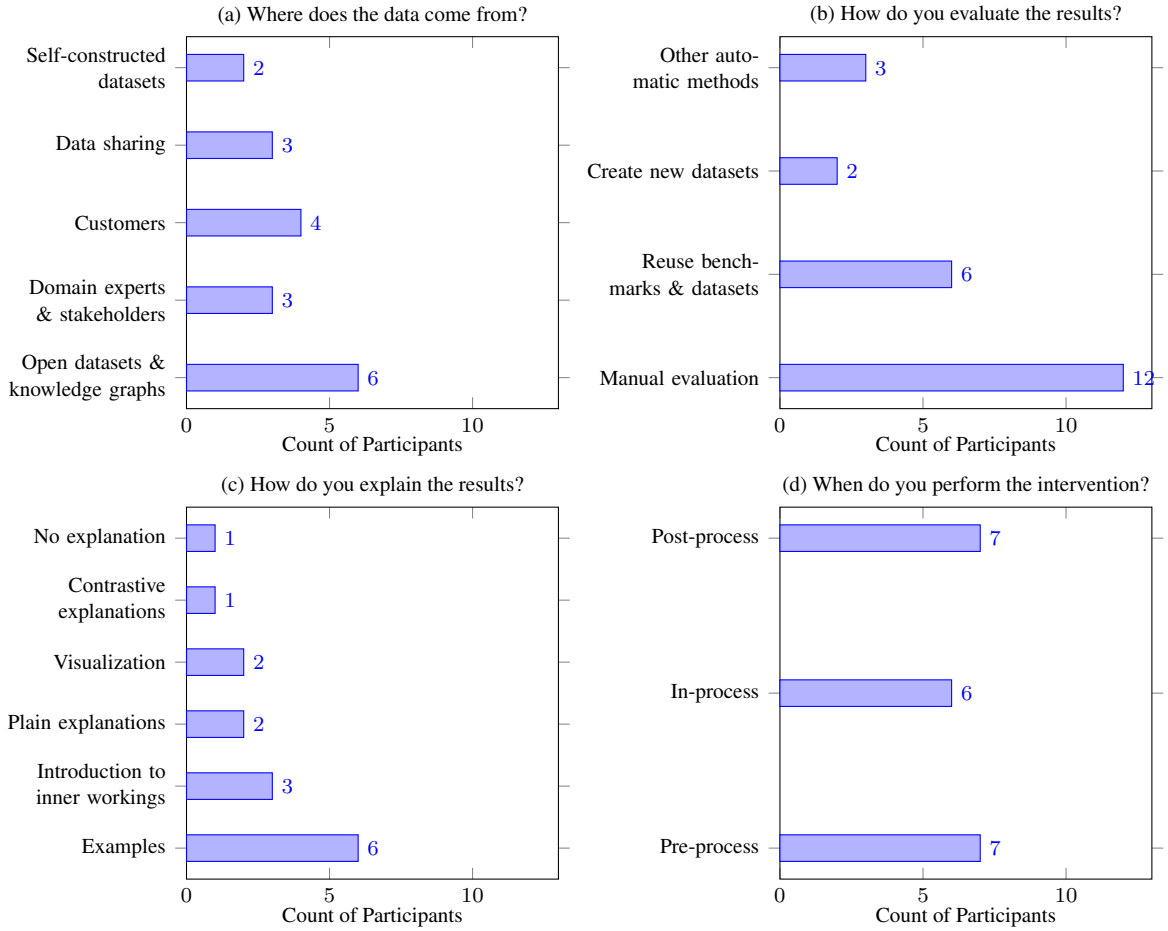

Fig. 5. Distribution of participant responses to four questions: (a) where does the data come from? (b) how do you evaluate the results? (c) how do you explain the results? (d) when do you perform the intervention? The x-axis represents the total number of participants (13), with multiple responses allowed per participant.

and reinforcement learning models that produce reasoning paths [145] to explain results. For generative models, asking them to generate additional or intermediate outputs, such as reasons for certain outputs could help. However, this is not always technically feasible. For example, adjusting embedding models to generate intermediate outputs for model understanding is not as straightforward as with LLMs through chain-of-thought [201]. Another approach proposed by participant *G* for understanding incorrect results is to seek training examples similar to some test data instances. This aligns with existing reviewed example-based explanations [188], such as [119], where similar training instances are returned as explanations for the assigned label of the candidate instance.

4.2.2. Data Provenance and Lineage

Do knowledge engineers know where the data comes from? **Among the 13 participants, 92.3% of them reported knowing the sources of their data. Data sources and providers varied, with participants often receiving multi-sourced data, depending on the projects they were working on.** As shown in Figure 5 (a), half of the participants used open knowledge graphs and publicly available data. However, this does not mean the data sources are always clear to them and verifiable. Not all participants are aware of how these datasets and knowledge graphs were created. For instance, participant *E* mentioned that they were unaware of how the benchmark datasets were created, but recognized that this data often has significant limitations, such as skewed distributions and incompleteness. Similarly, participant *H* noted that when using external APIs to obtain data, it was unclear where the data originated from. Data

can also be collected from domain experts and stakeholders or acquired from partners and collaborators with data sharing policies and platforms. However, participants using data from these sources reported similar challenges: some of them generally did not make extra efforts to understand the origins of the data and how exactly it was selected. Participant *M* also noted the difficulty in assessing the qualifications of annotators when data is manually annotated, as detailed information about their expertise is often unavailable. Participants from industry also collect data from customers, which can be sensitive and requires extra effort for data masking. A minority of participants constructed datasets themselves.

How do people keep track of data provenance and lineage? Among the participants flagging data provenance as essential, 69.2% actively tracked it in their tasks. Notably, all participants from industry and academia alike recognized the importance of data provenance and lineage and have established methods for documenting these aspects, given that their data primarily comes from partners and customers. The interview revealed a list of (semi-) automatic techniques either currently in use or planned for adoption to manage data provenance and lineage by the participants, including PROV Ontology[8], RDF Star[9], metadata, OpenRefine[10], Data Version Control (DVC)[11], data catalogs, NLP Interchange Format (NIF) [202], and blockchain. These tools document a set of details, such as the creation time, involved personnel, operation timelines, algorithms used to create the data, and potentially even the parameterization of these algorithms. Data provenance is tracked at different granular levels, from the model level (e.g., entire ontologies) to the data level (e.g., individual ontology elements). The availability of a wide range of tools offers knowledge engineers flexibility in fitting their specific pipelines. However, challenges and requirements remain. For instance, participant *M* noted the difficulty in determining the extent to which data provenance should be tracked and the level of details required. **There is also a preference for using standard representations based on standardized languages and vocabularies.** As participant *B* said, an ideal approach would involve "an explicit and standard representation of provenance and lineage directly attached to the produced artifacts", such as "metadata that accompanies the actual knowledge assets". It is also crucial to have an automatic, scalable, and trustworthy method for tracking data provenance and lineage, particularly when dealing with sensitive and frequently updated data. Participants also highlighted that integrating LLMs into the knowledge engineering process introduced new challenges for data provenance, as their extensive training data is often unknown.

4.2.3. Evaluation and Human Intervention

How do knowledge engineers evaluate the results? As shown in Figure 5(b), **most participants rely on human evaluation to evaluate the outcome of knowledge engineering tasks.** The human evaluation methods used are usually qualitative analysis or randomly sampling a subset of data for manual inspection. Depending on the task and the required expertise, the evaluators are usually domain experts and/or the researchers themselves. From our interviews, two participants reported recruiting domain experts for evaluation, one conducted evaluations both by the developers and domain experts, while the remaining 9 evaluated the results themselves. Several reasons contribute to the heavy reliance on human labor for evaluation. Firstly, there is a lack of available datasets and testing platforms. Secondly, existing datasets are often unsuitable for new scenarios. A model that performs well on current datasets may not necessarily perform well on new data, rendering the existing datasets less helpful. Thirdly, metrics used, such as average precision and accuracy, can sometimes be misleading. It is often the case that either the metrics look too good and the results are worse, or the metrics look very bad and the results are better than they appear. There are several additional issues with human evaluation. It is time-consuming, not scalable, and not always feasible. Additionally, randomly evaluating a subset of data might not accurately reflect the actual quality of the generated results. Participant *L* noted, "the main difficulty is to select an actual relevant sample".

Besides manual evaluation, 46.2% of participants also attempt to **reuse benchmarks and datasets** with ground truths to automate the evaluation process. They are aware that benchmarks are incomplete, biased, and skewed, and thus might not always truthfully reflect the models' abilities. Participant *M* from the industry mentioned that there are often gaps between the focus of the benchmarks and the requirements of customers. While benchmarks are more

[8] https://www.w3.org/TR/prov-o/
[9] https://w3c.github.io/rdf-star/
[10] https://openrefine.org
[11] https://dvc.org

focused on challenging cases and specific domains, customers are often interested in general domain cases, making the benchmarks less relevant to practical scenarios. Participants from academia also face difficulties, as mentioned by *G*, when there are insufficient resources for annotations or re-annotations, forcing them to work with the existing data. Participants also highlighted other methods to automate evaluation. Two participants mentioned using SHACL shapes for validation. Participant *J* stated they are developing a quality management concept and have a machine learning-based algorithm on top of other methods to estimate quality. Additionally, two participants indicated they **construct new datasets** for evaluating their methods. Although there are ongoing efforts to automate evaluation, the interviews revealed a consensus that different extents of human evaluation are always required.

What do people do when they find the results incorrect? Similar to human evaluation, the interviews revealed a consensus that human intervention is essential to compensate for the limitations of machines at various stages and levels of detail in the knowledge graph construction process. We categorized human intervention based on the stages at which it occurs, as shown in Figure 5(d). The first stage is **pre-processing**, where human intervention primarily involves working with the inputs. The most common approaches are data augmentation and cleaning. When working with LLMs, this also involves improving the input prompts. The second stage is **in-process intervention**, where researchers and knowledge engineers adjust the models and techniques or specific steps in the process to resolve issues. This could involve fine-tuning and re-training models, debugging code, or adding, removing, and modifying components and steps. Before making these modifications, there is typically a troubleshooting process to identify error patterns, systematic mistakes, and biases. Participant *B* mentioned that when using LLM-based pipelines, disambiguation is always a problem, so they either improve the prompts or add an extra disambiguation step. Another example, provided by participant *I*, is involving humans in identifying incorrect inference rules and then rerunning the models. The third stage involves directly modifying the model output, where humans manually correct a group of generated outputs (if the errors are manageable in size) or add post-hoc filters to exclude problematic results (i.e., **post-processing**). Statistically, 53.8% of participants adopted pre-processing methodologies, 46.2% engaged in in-process modifications, and 53.8% employed post-process corrections. This indicates that participants typically adopted multiple types of interventions at various stages. Moreover, the individuals performing the intervention are crucial. It's not just about their availability to check the results, but also about their expertise. As noted by participant *L*, when mistakes are "a mix of technical and domain-specific issues", it can be more challenging to identify and address them.

How do people explain to others their models and results? 92.3% of participants have experience explaining models and results to others. 38.5% of participants explained their models and results to stakeholders who may not have a technical background, typically domain experts. Eight participants explained their work with ontologists and knowledge engineers, who have a similar technical background, usually project partners and team members. Additionally, two participants mentioned producing explanations for educational purposes, targeting university students. This indicates that **designing and delivering explanations has become a crucial and challenging task in the knowledge graph lifecycle**. As highlighted by participant *L*, if the model performance does not meet stakeholder expectations and the model is not explainable, it does not foster acceptance or transparency.

The methods used for explanations are summarized in Figure 5(c). For now, there are no standardized methods for explaining the models and outputs in the knowledge graph lifecycle. **We observed that almost no methods from the literature review are used in participants' daily work scenarios.** We argue that there may be two main reasons for this. First, participants may not be aware of these methods and therefore rely on their own intuitive ways to explain results when needed. Second, the available methods may not be ready for practical use, and integrating existing XAI methods into their workflows is challenging. Only participant *B* mentioned having used one of the presented models in the example discussion session [136], finding it generally useful, although not all explanations produced by the model were helpful.

The most frequent method (used by six participants) is to select examples, including corner cases and errors, to explain the model's functionality, the relevance between input and output, the difficulties of the problems, and the range of the model's abilities. Three participants explained the pipelines and models through lectures and conceptual introductions to the technical components, often providing high-level overviews of the algorithms and models. The other two participants adopted visualization methods. Participant *A* reported success using visualizations to represent embeddings and clusters, which helped "define a clear boundary between technical and intuitive content".

One participant mentioned using contrastive explanations, such as why the machine made one decision instead of another. Two participants did not have a specific method but relied on plain explanations.

Using the same taxonomy adopted in the literature review in Section 3, we categorized the explanation methods collected from the interviews into two categories: contrastive explanations and example-based explanations as local post-hoc methods, and visualization, plain explanations, and introduction to inner workings as global post-hoc methods. Our analysis reveals that, out of the 14 responses regarding explanation methods, half of the responses are local post-hoc methods, while the other half are global post-hoc methods. Notably, no self-explaining methods were reported. In contrast, the literature review indicates that a substantial proportion of explainable methods consist of local self-explaining (59.5%) and local post-hoc (19%) methods. We posit that several factors contribute to this discrepancy. Self-explaining methods are preferred in academia-developed models because researchers often work on implementing models from scratch or improving models by adjusting components or integrating additional components for better performance. This objective aligns with the design of self-explaining models. Among the 50 local self-explaining papers reviewed, 37 pertain to link prediction models, which typically incorporate explanation mechanisms into their developed models, enhance existing models by making components explainable, or reformulate problems in an interpretable manner. For practitioners, however, implementing self-explaining methods poses challenges. Post-hoc explanations of model output offer greater flexibility, allowing practitioners to customize supporting evidence, visualize this evidence, and adapt explanations into other languages that are more comprehensible to their stakeholders.

Participants reported several challenges. **The most significant challenge when providing explanations is the gap in background, knowledge, and requirements between knowledge engineers/researchers and their stakeholders.** This gap makes it difficult to capture the interests and needs of stakeholders and to determine the appropriate level of technical detail for explanations. Reporting too many technical details may disinterest and frustrate stakeholders who lack a technical background. As participant *J* mentioned, "they might feel confused and show very little interest in what the numbers in the explanations represent." Participant *B* noted that there is often a gap in expectations, with knowledge engineers and researchers focusing on "understanding from the technical aspects" while stakeholders are more interested in "the practical use case and deployment perspectives." Participant *L* added that audiences often have "distorted expectations of the machine" and expect it to "reason or think as people do." Addressing this challenge involves finding a common language that is simple enough for non-technical stakeholders and customizing explanations for different audiences. Another challenge is generating robust and satisfactory explanations, which can be difficult due to a lack of ground truths, poor model performance, and the black-box nature of models.

4.3. Gaps and Challenges in Explainable KGC Solutions and Practical Usage

4.3.1. Use Cases from Interview Study

What are practical use cases of XAI models? We first compared the use cases in Section 3.1.2 with those collected from the interview study. We found that the use cases in Section 3.1.2 were largely reflected through the interview study, which also provided new insights and additional use cases. The most prominent use case, highlighted by 76.9% of participants, is understanding the model output and its inner workings. This includes providing supporting evidence, mapping results to the original input, and explaining how the models generate the output. This aligns with the previously identified use case of understanding performance and contributing factors. The second common use case, mentioned by 38.5% of participants, is debugging models and assisting in rectifying and adjusting them. This extends the previous use case of model debugging by indicating where the machine fails or is unstable, identifying systematic error patterns and problematic parts of data sets, and understanding mistakes and errors and their causes.

Two novel use cases were identified from the interviews. The first involves **enhancing human-machine interactions** by facilitating human involvement at various stages of the pipeline and providing effective interaction with the models. Participant *A* emphasized that having explainability can streamline the workflow, stating that "the more explainable the models are, the less human intervention is required" during model deployment. To this end, clear and informative explanations play a crucial role in bridging the knowledge gap among different stakeholders, ensuring a shared understanding at the right level. Additionally, simplifying the reuse and sharing of results and pipelines among stakeholders and other technical experts is crucial. This is similar to the model update use case

Task	Example 1			Example 2		
	Representation	Cite	Feedback	Representation	Cite	Feedback
Entity extraction	Words, attention score visualization	TMN [117]	⊕ ⊕ ⊕ ×	Training instances	Instance-based span [119]	⊕ ⊕ × ×
Relation extraction	Words	D-REX [121]	⊕ ⊕ × × ×	Logic rules	LogiRE [127]	✓ ✓ ⊕ ⊕ ×
Entity linking	Attribution scores and their visualization	LEMON [102]	× ×	Entity resolution rules	SystemER [183]	✓ ✓
Link prediction	Reasoning path	[136]	✓ ⊕	Training triples	Kelpie [95]	× ×
Inconsistency detection	Triples and their visualization	Abstraction [181]	× ×	/		

Table 8

User acceptance count of explainable examples, '✓' indicates vote for acceptable explanations, '×' indicates vote for unsatisfactory explanations, and '⊕' indicates vote for explanations that are somewhat reasonable but not fully trustworthy. For inconsistency detection, only one example is provided.

in Section 3.1.2, as reusing the pipeline in other processes also involves feeding new data into the pipeline and explaining the differences. By mapping the works discussed in the literature review to this use case, we found that human-in-the-loop approaches, including myDIG [187], SystemER [183], and TuneR [184], align perfectly with this context. Another novel use case that emerged from the interview study is **uncovering new and previously unnoticed insights**. This involves explaining how unexpected (but not necessarily wrong) results are obtained and offering additional details or information that may be overlooked when humans perform the same tasks. Among the works reviewed in 4.1.1, rule-based explanations, including those presented in [108, 129, 183, 184] for entity resolution and [154, 160] for link prediction, demonstrate notable potential for contributing to this use case.

4.3.2. XAI Example Discussion

Do current explainable solutions meet the requirements for practical use cases? During the example discussion session, participants provided feedback on various tasks: 5 commented on relation extraction, 4 on entity extraction, and 2 each on entity resolution, link prediction, and inconsistency detection (Table 8). **Overall, the examples seldom met participants' requirements, with only 17.9% of feedback responses being positive, while nearly half were negative.** Participants highlighted several concerns and issues regarding the practical adoption of the provided explanations. The primary issue, raised in 28.6% of responses, was that **the explanations were not sufficiently informative**. This could mean several things: the explanations might only cover one or a few aspects of the results, making them insufficient to fully explain the outcomes. Additionally, the correlation or relevance between the explanation and the output was often weak, rendering the explanations inadequate. For instance, using trigger words from the context to explain entity or relation extraction results might show some relevance but still fail to explain why the models produced those specific results instead of others. Participants also noted **the complexity of the explanations**, which made them difficult to understand and evaluate. A specific barrier was the use of technical terms. For example, explanations represented in logic rules were found useful but too complex for those without a technical background. Participant *M* mentioned that numerical thresholds used in rules were perplexing, and participant *C* expressed concerns that logic rules could quickly become overly complex, especially for long and intricate contexts. Once an error occurred, it was challenging to pinpoint its source, complicating the validation of the explanations. Additionally, visualizations or elements used in visualization-based explanations were not always clearly defined, such as numbers or color bars, and the representations were not in a standard language familiar to knowledge engineers or lay users. A third concern, raised by 21.4% of responses, was **the stability and coverage of the explainable models**. Participants questioned whether these models could consistently provide reliable explanations for all results. This issue, known as coverage, refers to how many cases from the entire set of results can be explained. Participants worried that there might not always be an explanation, or an explanation of sufficient quality, for all cases of interest. This concern was particularly focused on path-based and attention-based explanations. For path-based explanations, participants doubted the reliability of reasoning paths for every link prediction result. For attention-based explanations, they were concerned about the models' stability and the possibility of incorrect attention being paid to words, thus making the explanations less reliable.

4.4. Requirements for Explainable Approaches

What are characteristics of an explainable method that knowledge engineers and researchers expected? From the example discussions, we identified two key requirements. First, 30.8% of participants emphasized the need for a **confidence indicator** for models, explaining how confident the models are when producing results. This indicator should truthfully reflect the model's confidence in both the output and the explanations, and models should have the option to acknowledge uncertainty rather than provide incorrect answers or hallucinations. Wrong explanations, whether for correct or incorrect results, can bias users' impressions of the explanations, thereby eroding user trust in the model. Confidence indicators can reduce such cases and facilitate better human-machine collaboration by highlighting uncertain instances where human intervention is necessary. As participant *E* noted, when generating explanations for relation extraction, there should be "an option of like they don't know the relationships or they cannot predict this relationship based on the current context." Similarly, participant *F* stated, "we would like to make sure that the machine says it doesn't know when it doesn't know."

Secondly, the representation of explanations largely depends on the task and user. Although explanation formats like visualization and logic rules received varying levels of acceptance, the most acceptable representation for participants was **natural language**. 11 out of 13 participants preferred natural language explanations, either alone or combined with other representations such as visualization and logic rules. The main concern with visualizations and logic rules was semantic grounding, meaning the use of clearly defined language that ensures users understand the underlying semantics. Participants noted that visual notations without clear definitions are "prone to ambiguity". The same concern was raised for natural language explanations, as ambiguity can create challenges for human understanding of the model, thereby complicating human-machine interaction.

From the requirement elicitation questions, we also identified two common requirements from participants more directly. First, 30.8% of participants highlighted that the type of information users most require in explanations is **contextual information**. They want explainable models capable of tracing which inputs or intermediate outputs led to a certain result. As participant *D* mentioned, the ability to "pinpoint" refers to "identifying the minimal relevant information that a user needs to understand the result and the problem." This ability to map outputs to inputs can establish the basic trustworthiness of explainable models. Participant *I* added, "the sources from where the explanations are given are crucial to users." For example, in link prediction tasks, when the input is knowledge graphs, the required information includes which triples are used to derive the new one. Generally, this involves providing contextualized information with input data and knowledge graphs, reflecting the relevance between input and output, and the relevance between specific components or steps and the output.

Moreover, 30.8% of participants envisioned a solution involving a "hybrid pipeline" where people and machines work in cooperation, providing ways of **interaction** such as (1) machine-generated explanations for people to understand performance, corner or uncertain cases, etc.; and (2) a feedback loop for people to provide feedback on explanations to the system, explaining why something is right or wrong, or directly giving explanations, so that the machine can learn from this feedback and adjust itself. Follow that line, two participants specifically mentioned the need for iterative explanations in a conversational form. As participant *M* suggested, such "dynamic" explanations could extend over several rounds, allowing users to "keep asking for more depth if they see the need." Similarly, participant *G* believed that the explainable model should be able to select appropriate explanations based on the specific use case and input data, such as rules-based, similarity-based, or visualization-based explanations.

5. Explanation Design Blueprint

Based on the results from the literature review and interview study, we propose several guidelines and combine them into blueprints for designing explainable solutions in knowledge engineering tasks that are usable and trustworthy to target users, as shown in Figure 6.

The first step in designing explainable models involves XAI requirement analysis, which collects design insights and creates goals for explanations. Several factors must be carefully considered and investigated to capture the scope and objectives of explainable models.

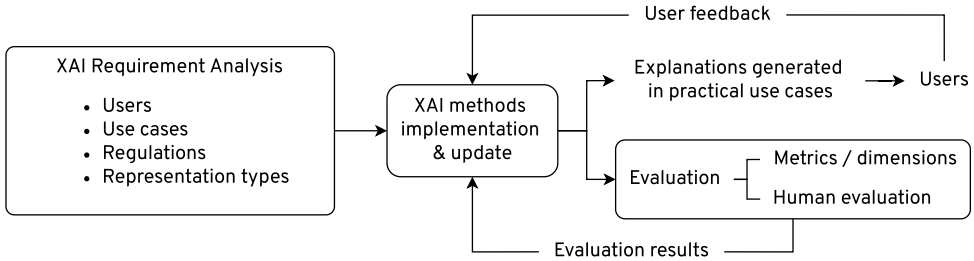

Fig. 6. The XAI methods design blueprint.

The most important factor is the **users** who consume the explanations. As participant *A* noted, "designing the system with the users in mind" and "users are the central component." In the context of the knowledge graph lifecycle, these users can be stakeholders, domain experts, knowledge engineers, etc. From the literature review, only 10 works explicitly described the intended users who engage with the generated explanations, as well as their background information. For instance, xERTE [139] reported the background information of respondents involved in the evaluation of explanations, including their education levels. TuneR [184] specifies that the tool is designed to support developers, including coders, scientists, and domain experts. And from the interview study, it is evident that the design of explanations should consider the users' level of understanding and interest in the technical details. Therefore, user analysis should include investigating their background, particularly their technical expertise and domain knowledge, and their expectations for the explanations, including the level of technical detail they require. Furthermore, if the consumers of explanations are involved in collaborative work with machines, understanding how they consume explanations and interact with models is crucial.

The second part of XAI requirement analysis focuses on the **use cases** of explanations. We identified six use cases, each requiring explanations to focus on different aspects. Developers need to decide which practical use cases the explanations will serve, which may extend beyond the six identified. For example, if explanations are used for model debugging and adjustment, they should provide details on inner workings and contributing factors to help identify error sources. A confidence indicator, as mentioned in Section 4.4, is also useful. If the goal is to enhance human-AI interactions, it is recommended to design mechanisms for providing explainability at multiple stages through collaboration and a good feedback loop to personalize the model's output based on user input.

The third factor is the **representation** of explanations, which primarily depends on the task and its related input and output (datatype, modality, or property) and user needs. For example, in knowledge extraction tasks where the input modality is text, the context of the original input might be useful (though not necessarily sufficient) as supporting evidence to show the relevance of the output. This needs to be combined with user analysis to determine what "language" the user speaks, such as description logic, natural language, images, etc.

Other factors, such as AI regulations mentioned in the Introduction, may also need to be considered in the requirement analysis. Moreover, this list of factors can be expanded based on actual scenarios. Such requirement analysis guides the selection and implementation of XAI methods, ensuring they meet the requirements of practical use cases. After implementing XAI methods, the workflow involves iterative loops for maintaining and continuously improving the methods. One loop focuses on the evaluation and assessment of explanations [203]. Evaluation methods should go beyond anecdotal evidence, selecting appropriate metrics or designing evaluation paradigms. Another iterative loop, derived from the "hybrid pipeline" requirements in Section 4.4, aims to improve explainable models and explanations in practical scenarios. Users who consume the explanations provide feedback and example explanations, which can be used in various ways to enhance the XAI model. This includes creating datasets of explanations for training and fine-tuning XAI models, providing few-shot examples, or even abstracting improvement directions for architecture-level adjustments.

6. Conclusion and Future Work

6.1. Conclusion

In this paper, we adopted a mixed methodology, conducting a literature review on explainable methods within the domain of knowledge graph construction and an interview study on the same topic with 13 participants to capture how XAI methods support knowledge engineering. We performed the analysis in three dimensions, tasks related to knowledge graph construction, the taxonomy of XAI methods, and the use cases of XAI methods in knowledge graph construction. We observed that the most effort has been directed towards automation and explainability in entity extraction, relation extraction, entity linking, and link prediction. Additionally, we considered the use cases in explainable automatic knowledge graph construction, such as ML model selecting and building, ML model debugging, understanding performance and contributing factors, and managing updates. The interview study largely corroborated the considered use cases, adding new insights and highlighting additional use cases, including enhancing human-machine interactions and providing new insights from unexpected results. We found that the reviewed models primarily focused on explaining the performance and contributing factors to the outcome while neglecting other use cases, such as error detection and correction, which could help establish trust with users. The interview study revealed that while current knowledge engineering models and techniques exhibit varying degrees of automation and understanding, significant challenges remain in data provenance, evaluation methods, and providing clear explanations to stakeholders. The current explainable solutions often fell short of participants' requirements, with concerns about their informativeness, complexity, and reliability. These insights established a foundational understanding of critical transparency factors, enabling the development of a blueprint for designing explainable methods for knowledge engineering tasks.

6.2. Future Work

We identified five future directions for research on explainable automatic knowledge graph construction. First, going back to prior literature on knowledge engineering methodologies [57, 58, 204, 205], there are many **tasks and activities where automation remains an exception**. Aside from the tasks in Figure 3, there is an opportunity to think about other ways for AI assistance to add value: for instance, one design principle of KGs is that they are meant to integrate across multiple sources and be able to tackle evolving requirements. Reusing existing schemas or ontologies can help with interoperability, but the task of finding or assessing an ontology for reuse is still mostly manual. At the other end of the lifecycle, documenting KGs can help with maintenance and reuse, and advances in generative AI make it a chief candidate for automation. While we found a range of explainable link prediction approaches, it would be useful to dive deeper into this sub-field to understand the extent to which these different approaches solve common concerns around the quality of KGs. One difference between representing knowledge in a KG and a machine-learning model is that a KG can provide guarantees about the validity of the information, its provenance, its currency, etc. upon retrieval. However, this is predicated by KGs being regularly audited according to these and other quality dimensions and improved. Link prediction is one way to do this, alongside many others, e.g., debiasing [22]. Furthermore, while knowledge acquisition is generally well represented in the literature, a lot of work focuses on text rather than other data modalities, which is a concern in many KG application areas, e.g., enterprise data management (which needs to work with structured data) or cultural heritage (where a lot of domain data is neither text nor numbers).

Second, as we noted earlier, the fewest of approaches look at **the human-in-the-loop aspects of KG construction**, including human agency and oversight, feedback, etc [192] and **the integration of the developed models into established knowledge-engineering practices**. While there is a lot of work in human-AI interaction and interactive ML in the HCI community, they tend to focus so far on simpler ML models and different applications that the knowledge production scenarios we are interested in. One exception is the work on ORES [206], a participatory ML system used in Wikipedia and Wikidata (a large open-source KG). However, the Wikidata KG construction process is unique because it is community-based, with more than 24K active contributors[12] who receive AI assistance

[12] https://www.wikidata.org/wiki/Wikidata:Statistics

for distinct tasks such as vandalism detection and consistency checks. We need to follow their example to develop the same types of workflows and techniques for other KG construction scenarios - in most cases, these involve much smaller teams and different tool environments. The majority of existing integrated development environments (IDEs) for KGs (e.g., PoolParty[13], data.world[14], Protégé[15]) assume KGs are mostly built manually, with some basic automation to speed-up routine tasks like translating node labels or creating documentation from node and edge descriptions. LLMs offer chances to develop novel KG editing tools and interactions, allowing people to interact with their AI agents via natural language and ensuring transparency. Meanwhile, developers working with KGs require KG-related process blueprints that utilize AI algorithms and adhere to AI regulations for creating downstream applications. Recent works, such as OntoChat [63], which incorporate chatbots into tasks during the requirement engineering stage of ontology engineering, have introduced new directions for human-in-the-loop practices, driven by AI chatbots and agents.

Thirdly, our research flagged the need for **better evaluations on explanations**, which encompasses metrics, benchmarks, and datasets, as well as toolkits and guidance for conducting studies that assess how effective the explanations supplied in KG construction tasks are as proxies and enablers for transparent and hence trusted KGs.

Fourthly, our research revealed an imbalance in the distribution of use cases identified in the study. There was a strong emphasis on understanding the inner workings, performance, and contributing factors of models, while relatively few efforts were made to address other use cases also demanded by the community, such as model debugging, model updating, and human-AI interaction. However, our example discussions indicated that the reviewed explanations often failed to meet these requirements, and participants expressed low confidence in using them in their work or providing them to users. A future direction, reflected in our study and requested by the community, involves **adapting current explainable methods to representations and formats that are reusable across multiple use cases**.

Finally, although our research provided a blueprint for designing XAI methods, **practical applications and verification of the blueprint** are missing. Given the different use cases and groups of stakeholders in a knowledge engineering project, several details can be enriched. For instance, parts of the blueprint, such as the user feedback loop, can be refined. Future work could investigate what formats, workflows, and feedback frequencies best prompt users to provide high-quality explanations efficiently.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J.E.L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.N. Ngomo, S.M. Rashid, A. Rula, L. Schmelzeisen, J.F. Sequeda, S. Staab and A. Zimmermann, Knowledge Graphs, *CoRR* **abs/2003.02320** (2020). https://arxiv.org/abs/2003.02320.

[2] C. Peng, F. Xia, M. Naseriparsa and F. Osborne, Knowledge Graphs: Opportunities and Challenges, *Artificial Intelligence Review* (2023), 1–32. doi:10.1007/s10462-023-10465-9.

[3] J. Chen, Y. Geng, Z. Chen, J.Z. Pan, Y. He, W. Zhang, I. Horrocks and H. Chen, Zero-shot and Few-shot Learning with Knowledge Graphs: A Comprehensive Survey, *CoRR* **abs/2112.10006** (2021). https://arxiv.org/abs/2112.10006.

[4] P.S.H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel and D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, *CoRR* **abs/2005.11401** (2020). https://arxiv.org/abs/2005.11401.

[5] L. Yang, H. Chen, Z. Li, X. Ding and X. Wu, Give us the Facts: Enhancing Large Language Models With Knowledge Graphs for Fact-Aware Language Modeling, *IEEE Trans. on Knowl. and Data Eng.* **36**(7) (2024), 3091–3110–. doi:10.1109/TKDE.2024.3360454.

[6] I. Tiddi and S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, *Artificial Intelligence* **302** (2022), 103627. doi:https://doi.org/10.1016/j.artint.2021.103627. https://www.sciencedirect.com/science/article/pii/S0004370221001788.

[7] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong and Q. He, A Survey on Knowledge Graph-Based Recommender Systems, *IEEE Transactions on Knowledge and Data Engineering* **34**(08) (2022), 3549–3568. doi:10.1109/TKDE.2020.3028705.

[8] J. Sequeda and O. Lassila, *Designing and Building Enterprise Knowledge Graphs*, Springer International Publishing, 2021. ISSN 2691-2031. ISBN 9783031019166. doi:10.1007/978-3-031-01916-6.

[13] https://www.poolparty.biz/

[14] https://data.world/

[15] https://protege.stanford.edu/

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, Language Models are Unsupervised Multitask Learners, *OpenAI* (2019), Accessed: 2024-11-15. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[10] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, Language Models are Few-Shot Learners, *CoRR* **abs/2005.14165** (2020). https://arxiv.org/abs/2005.14165.

[11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, LLaMA: Open and Efficient Foundation Language Models, *CoRR* **abs/2302.13971** (2023). doi:10.48550/ARXIV.2302.13971. https://doi.org/10.48550/arXiv.2302.13971.

[12] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P.S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X.E. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov and T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, *CoRR* **abs/2307.09288** (2023). doi:10.48550/ARXIV.2307.09288. https://doi.org/10.48550/arXiv.2307.09288.

[13] F. Petroni, T. Rocktäschel, P.S.H. Lewis, A. Bakhtin, Y. Wu, A.H. Miller and S. Riedel, Language Models as Knowledge Bases?, *CoRR* **abs/1909.01066** (2019). http://arxiv.org/abs/1909.01066.

[14] S. Razniewski, A. Yates, N. Kassner and G. Weikum, Language Models As or For Knowledge Bases, *CoRR* **abs/2110.04888** (2021). https://arxiv.org/abs/2110.04888.

[15] J.Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, R. Biswas, G. de Melo, A. Bonifati, E. Vakaj, M. Dragoni and D. Graux, Large Language Models and Knowledge Graphs: Opportunities and Challenges, *Transactions on Graph Data and Knowledge* **1**(1) (2023), 2:1–2:38. doi:10.4230/TGDK.1.1.2.

[16] G. Weikum, L. Dong, S. Razniewski and F.M. Suchanek, Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases, *CoRR* **abs/2009.11564** (2020). https://arxiv.org/abs/2009.11564.

[17] A. Hur, N. Janjua and M. Ahmed, A Survey on State-of-the-art Techniques for Knowledge Graphs Construction and Challenges ahead, in: *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2021, pp. 99–103. doi:10.1109/AIKE52691.2021.00021.

[18] G.e. Tamašauskaitė and P. Groth, Defining a Knowledge Graph Development Process Through a Systematic Review, *ACM Trans. Softw. Eng. Methodol.* **32**(1) (2023). doi:10.1145/3522586.

[19] M. Hofer, D. Obraczka, A. Saeedi, H. Köpcke and E. Rahm, Construction of Knowledge Graphs: Current State and Challenges, *Information* **15**(8) (2024). doi:10.3390/info15080509. https://www.mdpi.com/2078-2489/15/8/509.

[20] C.T. Wolf, From Knowledge Graphs to Knowledge Practices: On the Need for Transparency and Explainability in Enterprise Knowledge Graph Applications, in: *Knowledge Graph Bias Workshop*, 2020. https://api.semanticscholar.org/CorpusID:230516430.

[21] D. Abián, A. Meroño-Pe nuela and E. Simperl, An Analysis of Content Gaps Versus User Needs in the Wikidata Knowledge Graph, in: *The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings*, Springer-Verlag, Berlin, Heidelberg, 2022, pp. 354–374–. ISBN 978-3-031-19432-0. doi:10.1007/978-3-031-19433-7_21.

[22] J. Fisher, A. Mittal, D. Palfrey and C. Christodoulopoulos, Debiasing knowledge graph embeddings, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He and Y. Liu, eds, Association for Computational Linguistics, Online, 2020, pp. 7332–7345. doi:10.18653/v1/2020.emnlp-main.595. https://aclanthology.org/2020.emnlp-main.595/.

[23] B. Zhang, A. Meroño Peñuela and E. Simperl, Towards Explainable Automatic Knowledge Graph Construction with Human-in-the- Loop, in: *HHAI 2023: Augmenting Human Intellect*, IOS Press, 2023, pp. 274–289. doi:10.3233/FAIA230091.

[24] P. Groth, E. Simperl, M. van Erp and D. Vrandečić, Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century (Dagstuhl Seminar 22372), *Dagstuhl Reports* **12**(9) (2023), 60–120. doi:10.4230/DagRep.12.9.60.

[25] J.D. Lee and K.A. See, Trust in Automation: Designing for Appropriate Reliance, *Human Factors* **46**(1) (2004), 50–80, PMID: 15151155. doi:10.1518/hfes.46.1.50_30392. https://doi.org/10.1518/hfes.46.1.50_30392.

[26] D. Vrandečić and M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, *Commun. ACM* **57**(10) (2014), 78–85–. doi:10.1145/2629489.

[27] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2) (2015), 167–195. doi:10.3233/SW-140134.

[28] T. Pellissier Tanon, G. Weikum and F. Suchanek, YAGO 4: A Reason-able Knowledge Base, in: *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A.L. Gentile, P. Haase and M. Cochez, eds, Springer International Publishing, Cham, 2020, pp. 583–596. ISBN 978-3-030-49461-2.

[29] R. Speer, J. Chin and C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, *CoRR* **abs/1612.03975** (2016). http://arxiv.org/abs/1612.03975.

[30] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali and A. ten Teije, Modular design patterns for hybrid learning and reasoning systems, *Appl. Intell.* **51**(9) (2021), 6528–6546. doi:10.1007/S10489-021-02394-3. https://doi.org/10.1007/s10489-021-02394-3.

[31] A. Breit, L. Waltersdorfer, F.J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A.T. Teije and F. Van Harmelen, Combining Machine Learning and Semantic Web: A Systematic Mapping Study, *ACM Comput. Surv.* **55**(14s) (2023). doi:10.1145/3586163.

[32] F. Poursabzi-Sangdeh, D.G. Goldstein, J.M. Hofman, J.W. Vaughan and H.M. Wallach, Manipulating and Measuring Model Interpretability, *CoRR* **abs/1802.07810** (2018). http://arxiv.org/abs/1802.07810.

[33] A. Smith-Renner, R. Fan, M. Birchfield, T. Wu, J. Boyd-Graber, D.S. Weld and L. Findlater, No Explainability without Accountability: An Empirical Study of Explanations and Feedback in Interactive ML, in: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1–13–. ISBN 9781450367080. doi:10.1145/3313831.3376624.

[34] X. Wang and M. Yin, Effects of Explanations in AI-Assisted Decision Making: Principles and Comparisons **12**(4) (2022). doi:10.1145/3519266.

[35] U. Ehsan, Q.V. Liao, M. Muller, M.O. Riedl and J.D. Weisz, Expanding Explainability: Towards Social Transparency in AI systems, in: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450380966. doi:10.1145/3411764.3445188.

[36] D. Kaur, S. Uslu, K.J. Rittichier and A. Durresi, Trustworthy Artificial Intelligence: A Review, *ACM Comput. Surv.* **55**(2) (2022). doi:10.1145/3491209.

[37] S. Larsson and F. Heintz, Transparency in artificial intelligence, *Internet Policy Review* **9**(2) (2020). doi:10.14763/2020.2.1469.

[38] G. Schwalbe and B. Finzel, XAI Method Properties: A (Meta-)study, *CoRR* **abs/2105.07190** (2021). https://arxiv.org/abs/2105.07190.

[39] M.T. Ribeiro, S. Singh and C. Guestrin, 'Why Should I Trust You?': Explaining the Predictions of Any Classifier, *CoRR* **abs/1602.04938** (2016). http://arxiv.org/abs/1602.04938.

[40] S.M. Lundberg and S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: *Advances in Neural Information Processing Systems 30*, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, Curran Associates, Inc., 2017, pp. 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[41] P. Hase and M. Bansal, Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, 2020, pp. 5540–5552. doi:10.18653/v1/2020.acl-main.491. https://aclanthology.org/2020.acl-main.491.

[42] D. Minh, H.X. Wang, Y.F. Li and T.N. Nguyen, Explainable Artificial Intelligence: A Comprehensive Review, *Artif. Intell. Rev.* **55**(5) (2022), 3503–3568–. doi:10.1007/s10462-021-10088-y.

[43] S. Mohseni, N. Zarei and E.D. Ragan, A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems, *ACM Trans. Interact. Intell. Syst.* **11**(3–4) (2021). doi:10.1145/3387166.

[44] G. Vilone and L. Longo, Explainable Artificial Intelligence: a Systematic Review, *CoRR* **abs/2006.00093** (2020). https://arxiv.org/abs/2006.00093.

[45] A.B. Arrieta, N.D. Rodríguez, J.D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI, *CoRR* **abs/1910.10045** (2019). http://arxiv.org/abs/1910.10045.

[46] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas and P. Sen, A Survey of the State of Explainable AI for Natural Language Processing, *CoRR* **abs/2010.00711** (2020). https://arxiv.org/abs/2010.00711.

[47] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *Artificial Intelligence* **267** (2019), 1–38. doi:https://doi.org/10.1016/j.artint.2018.07.007. https://www.sciencedirect.com/science/article/pii/S0004370218305988.

[48] Y. Rong, T. Leemann, T.-T. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci and E. Kasneci, Towards Human-Centered Explainable AI: A Survey of User Studies for Model Explanations, *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(4) (2024), 2104–2122–. doi:10.1109/TPAMI.2023.3331846.

[49] B. Mittelstadt, C. Russell and S. Wachter, Explaining Explanations in AI, in: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 279–288–. ISBN 9781450361255. doi:10.1145/3287560.3287574.

[50] A.D. Preece, D. Harborne, D. Braines, R. Tomsett and S. Chakraborty, Stakeholders in Explainable AI, *CoRR* **abs/1810.00184** (2018). http://arxiv.org/abs/1810.00184.

[51] G. Ras, M. van Gerven and P. Haselager, *Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges*, in: *Explainable and Interpretable Models in Computer Vision and Machine Learning*, H.J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü and M. van Gerven, eds, Springer International Publishing, Cham, 2018, pp. 19–36. ISBN 978-3-319-98131-4. doi:10.1007/978-3-319-98131-4_2.

[52] Q.V. Liao, D. Gruen and S. Miller, Questioning the AI: Informing Design Practices for Explainable AI User Experiences, in: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1–15–. ISBN 9781450367080. doi:10.1145/3313831.3376590.

[53] S. Dhanorkar, C.T. Wolf, K. Qian, A. Xu, L. Popa and Y. Li, Who Needs to Know What, When?: Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle, in: *Designing Interactive Systems Conference 2021*, DIS '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1591–1602–. ISBN 9781450384766. doi:10.1145/3461778.3462131.

[54] S.S.Y. Kim, E.A. Watkins, O. Russakovsky, R. Fong and A. Monroy-Hernández, "Help Me Help the AI": Understanding How Explainability Can Support Human-AI Interaction, in: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, Association for Computing Machinery, New York, NY, USA, 2023. ISBN 9781450394215. doi:10.1145/3544548.3581001.

[55] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen and C. Seifert, From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI, *CoRR* **abs/2201.08164** (2022). https://arxiv.org/abs/2201.08164.

[56] M. Chromik and M. Schuessler, A Taxonomy for Human Subject Evaluation of Black-Box Explanations in XAI, *ExSS-ATEC@IUI* **1** (2020). https://ceur-ws.org/Vol-2582/paper9.pdf.

[57] G. Schreiber, *Knowledge Engineering and Management: The CommonKADS Methodology*, A Bradford book, MIT Press, 2000. ISBN 9780262193009. https://books.google.co.uk/books?id=HlXOW_1fsIEC.

[58] R. Studer, V.R. Benjamins and D. Fensel, Knowledge engineering: Principles and methods, *Data & Knowledge Engineering* **25**(1) (1998), 161–197. doi:https://doi.org/10.1016/S0169-023X(97)00056-6. https://www.sciencedirect.com/science/article/pii/S0169023X97000566.

[59] D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich and A. Wahler, *Knowledge Graphs*, Springer, 2020. doi:10.1007/978-3-030-37439-6.

[60] M.K. Sarker, L. Zhou, A. Eberhart and P. Hitzler, Neuro-symbolic Artificial Intelligence, *AI Commun.* **34**(3) (2021), 197–209–. doi:10.3233/AIC-210084.

[61] P. Schneider, T. Schopf, J. Vladika, M. Galkin, E. Simperl and F. Matthes, A Decade of Knowledge Graphs in Natural Language Processing: A Survey, in: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online only, 2022, pp. 601–614. https://aclanthology.org/2022.aacl-main.46.

[62] H. Ye, N. Zhang, H. Chen and H. Chen, Generative Knowledge Graph Construction: A Review, *CoRR* **abs/2210.12714** (2022). doi:10.48550/arXiv.2210.12714.

[63] B. Zhang, V.A. Carriero, K. Schreiberhuber, S. Tsaneva, L.S. González, J. Kim and J. de Berardinis, OntoChat: A Framework for Conversational Ontology Engineering Using Language Models, in: *European Semantic Web Conference*, Springer, 2024, pp. 102–121. doi:10.1007/978-3-031-78952-6_10.

[64] E. Simperl and M. Luczak-Rösch, Collaborative ontology engineering: a survey, *The Knowledge Engineering Review* **29**(1) (2014), 101–131–. doi:10.1017/S0269888913000192.

[65] U. Simsek, E. Kärle, K. Angele, E. Huaman, J. Opdenplatz, D. Sommer, J. Umbrich and D. Fensel, A Knowledge Graph Perspective on Knowledge Engineering, *SN Comput. Sci.* **4**(1) (2022). doi:10.1007/s42979-022-01429-x.

[66] F. van Harmelen and A. ten Teije, A Boxology of Design Patterns for Hybrid Learning and Reasoning Systems, *Journal of Web Engineering* **18**(1–3) (2019). https://journals.riverpublishers.com/index.php/JWE/article/view/3175.

[67] H.F. Witschel, C. Pande, A. Martin, E. Laurenzi and K. Hinkelmann, *Visualization of Patterns for Hybrid Learning and Reasoning with Human Involvement*, in: *New Trends in Business Information Systems and Technology: Digital Innovation and Digital Business Transformation*, R. Dornberger, ed., Springer International Publishing, Cham, 2021, pp. 193–204. ISBN 978-3-030-48332-6. doi:10.1007/978-3-030-48332-6_13.

[68] C.W. Holsapple and K.D. Joshi, A Collaborative Approach to Ontology Design, *Commun. ACM* **45**(2) (2002), 42–47–. doi:10.1145/503124.503147.

[69] S. Auer and H. Herre, RapidOWL — An Agile Knowledge Engineering Methodology, in: *Perspectives of Systems Informatics*, I. Virbitskaite and A. Voronkov, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 424–430. ISBN 978-3-540-70881-0. doi:10.1007/978-3-540-70881-0_36.

[70] S. Braun, A.P. Schmidt, A. Walter, G. Nagypál and V. Zacharias, Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering, in: *CKC*, 2007. https://ceur-ws.org/Vol-273/paper_14.pdf.

[71] C. Debruyne, T.-K. Tran and R. Meersman, Grounding Ontologies with Social Processes and Natural Language, *Journal on Data Semantics* **2**(2–3) (2013), 89–118. doi:10.1007/s13740-013-0023-3.

[72] K. Kotis and A. Vouros, Human-Centered Ontology Engineering: The HCOME Methodology, *Knowl. Inf. Syst.* **10**(1) (2006), 109–131–. doi:10.1007/s10115-005-0227-4.

[73] D. Vrandečić, H.S. Pinto, C. Tempich and Y. Sure-Vetter, The DILIGENT knowledge processes, *Journal of Knowledge Management* **9** (2005), 85–96. doi:10.1108/13673270510622474.

[74] A. de Moor, P. De Leenheer and R. Meersman, DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering, in: *Conceptual Structures: Inspiration and Application*, H. Schärfe, P. Hitzler and P. Øhrstrøm, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 189–202. ISBN 978-3-540-35902-9. doi:10.1007/11787181_14.

[75] N. Guarino and C.A. Welty, *An Overview of OntoClean*, in: *Handbook on Ontologies*, S. Staab and R. Studer, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 151–171. ISBN 978-3-540-24750-0. doi:10.1007/978-3-540-24750-0_8.

[76] M. Poveda-Villalón, A. Gómez-Pérez and M.C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-Line Tool for Ontology Evaluation, *Int. J. Semant. Web Inf. Syst.* **10**(2) (2014), 7–34–. doi:10.4018/ijswis.2014040102.

[77] E.F. Kendall and D.L. McGuinness, *Requirements and Use Cases*, in: *Ontology Engineering*, Springer International Publishing, Cham, 2019, pp. 25–44. ISBN 978-3-031-79486-5. doi:10.1007/978-3-031-79486-5_3.

[78] V. Yadav and S. Bethard, A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, *CoRR* **abs/1910.11470** (2019). http://arxiv.org/abs/1910.11470.

[79] Y. Lin, S. Shen, Z. Liu, H. Luan and M. Sun, Neural Relation Extraction with Selective Attention over Instances, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2124–2133. doi:10.18653/v1/P16-1200. https://aclanthology.org/P16-1200.

[80] Ö. Sevgili, A. Shelmanov, M.Y. Arkhipov, A. Panchenko and C. Biemann, Neural Entity Linking: A Survey of Models based on Deep Learning, *CoRR* **abs/2006.00575** (2020). https://arxiv.org/abs/2006.00575.

[81] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer and J. Lehmann, Crowdsourcing Linked Data Quality Assessment, in: *The Semantic Web – ISWC 2013*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J.X. Parreira, L. Aroyo, N. Noy, C. Welty and K. Janowicz, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 260–276. ISBN 978-3-642-41338-4. doi:10.1007/978-3-642-41338-4_17.

[82] A. Revenko, M. Sabou, A. Ahmeti and M. Schauer, Crowd-Sourced Knowledge Graph Extension: A Belief Revision Based Approach, in: *Proceedings of the HCOMP 2018 Works in Progress and Demonstration Papers Track of the sixth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2018), Zurich, Switzerland, July 5-8, 2018*, A. Bozzon and M. Venanzi, eds, CEUR Workshop Proceedings, Vol. 2173, CEUR-WS.org, 2018. https://ceur-ws.org/Vol-2173/paper4.pdf.

[83] Z. Kou, Y. Zhang, D. Zhang and D. Wang, CrowdGraph: A Crowdsourcing Multi-modal Knowledge Graph Approach to Explainable Fauxtography Detection, *Proc. ACM Hum.-Comput. Interact.* **6**(CSCW2) (2022). doi:10.1145/3555178.

[84] A. Piscopo and E. Simperl, What we talk about when we talk about wikidata quality: a literature survey, in: *Proceedings of the 15th International Symposium on Open Collaboration*, OpenSym '19, Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450363198. doi:10.1145/3306446.3340822.

[85] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe and P.A. Szekely, A Study of the Quality of Wikidata, *CoRR* **abs/2107.00156** (2021). https://arxiv.org/abs/2107.00156.

[86] E. Koutsiana, T. Yadav, N. Jain, A. Meroño-Peñuela and E. Simperl, Agreeing and Disagreeing in Collaborative Knowledge Graph Construction: An Analysis of Wikidata, *CoRR* **abs/2306.11766** (2023). doi:10.48550/ARXIV.2306.11766. https://doi.org/10.48550/arXiv.2306.11766.

[87] R. Qarout, A. Checco, G. Demartini and K. Bontcheva, Platform-Related Factors in Repeatability and Reproducibility of Crowdsourcing Tasks, *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* **7**(1) (2019), 135–143. doi:10.1609/hcomp.v7i1.5264. https://ojs.aaai.org/index.php/HCOMP/article/view/5264.

[88] P. Cimiano and H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semant. Web* **8**(3) (2017), 489–508–. doi:10.3233/SW-160218.

[89] A. Rossi, D. Firmani, A. Matinata, P. Merialdo and D. Barbosa, Knowledge Graph Embedding for Link Prediction: A Comparative Analysis, *CoRR* **abs/2002.00819** (2020). https://arxiv.org/abs/2002.00819.

[90] M. Zamini, H. Reza and M. Rabiei, A Review of Knowledge Graph Completion, *Information* **13**(8) (2022). doi:10.3390/info13080396. https://www.mdpi.com/2078-2489/13/8/396.

[91] M. Wiegmann, M. Völske, B. Stein and M. Potthast, Language Models as Context-sensitive Word Search Engines, in: *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 39–45. doi:10.18653/v1/2022.in2writing-1.5. https://aclanthology.org/2022.in2writing-1.5.

[92] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A.M. Dai, T.S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov and N. Fiedel, PaLM: Scaling Language Modeling with Pathways, arXiv, 2022. doi:10.48550/ARXIV.2204.02311. https://arxiv.org/abs/2204.02311.

[93] J. Guo, J. Li, D. Li, A.M.H. Tiong, B. Li, D. Tao and S.C.H. Hoi, From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models, arXiv, 2022. doi:10.48550/ARXIV.2212.10846. https://arxiv.org/abs/2212.10846.

[94] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang and H. Wang, Retrieval-Augmented Generation for Large Language Models: A Survey, *CoRR* **abs/2312.10997** (2023). doi:10.48550/ARXIV.2312.10997. https://doi.org/10.48550/arXiv.2312.10997.

[95] A. Rossi, D. Firmani, P. Merialdo and T. Teofili, Explaining Link Prediction Systems Based on Knowledge Graph Embeddings, in: *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2062–2075–. ISBN 9781450392495. doi:10.1145/3514221.3517887.

[96] F. Bianchi, G. Rossiello, L. Costabello, M. Palmonari and P. Minervini, Knowledge Graph Embeddings and Explainable AI, *CoRR* **abs/2004.14843** (2020). https://arxiv.org/abs/2004.14843.

[97] M.J. Page, D. Moher, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan, R. Chou, J. Glanville, J.M. Grimshaw, A. Hróbjartsson, M.M. Lalu, T. Li, E.W. Loder, E. Mayo-Wilson, S. McDonald, L.A. McGuinness, L.A. Stewart, J. Thomas, A.C. Tricco, V.A. Welch, P. Whiting and J.E. McKenzie, PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews, *BMJ* **372** (2021). doi:10.1136/bmj.n160. https://www.bmj.com/content/372/bmj.n160.

[98] K. Amarasinghe, K.T. Rodolfa, H. Lamba and R. Ghani, Explainable machine learning for public policy: Use cases, gaps, and research directions, *Data & Policy* **5** (2023), e5. doi:10.1017/dap.2023.2.

[99] A. Adhikari, E. Wenink, J. van der Waa, C. Bouter, I. Tolios and S. Raaijmakers, Towards FAIR Explainable AI: a standardized ontology for mapping XAI solutions to use cases, explanations, and AI systems, in: *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 562–568–. ISBN 9781450396318. doi:10.1145/3529190.3535693.

[100] R. Confalonieri, O. Kutz, D. Calvanese, J.M. Alonso, S.-M. Zhou, S. Chari, O. Seneviratne, M. Ghalwash, S. Shirai, D.M. Gruen, P. Meyer, P. Chakraborty and D.L. McGuinness, Explanation Ontology: A general-purpose, semantic representation for supporting user-centered explanations, *Semantic Web* **15**(4) (2024), 959–989. doi:10.3233/SW-233282.

[101] D. Firmani, L. Tanca and R. Torlone, Ethical Dimensions for Data Quality, *J. Data and Information Quality* **12**(1) (2019). doi:10.1145/3362121.
[102] N. Barlaug, LEMON: Explainable Entity Matching, *CoRR* **abs/2110.00516** (2021). https://arxiv.org/abs/2110.00516.
[103] J. Wang and Y. Li, Minun: Evaluating Counterfactual Explanations for Entity Matching, in: *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning*, DEEM '22, Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450393751. doi:10.1145/3533028.3533304.
[104] A. Baraldi, F. Del Buono, M. Paganelli and F. Guerra, Landmark Explanation: An Explainer for Entity Matching Models, in: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4680–4684–. ISBN 9781450384469. doi:10.1145/3459637.3481981.
[105] T. Teofili, D. Firmani, N. Koudas, V. Martello, P. Merialdo and D. Srivastava, Effective Explanations for Entity Resolution Models, arXiv, 2022. doi:10.48550/ARXIV.2203.12978. https://arxiv.org/abs/2203.12978.
[106] V. Di Cicco, D. Firmani, N. Koudas, P. Merialdo and D. Srivastava, Interpreting Deep Learning Models for Entity Resolution: An Experience Report Using LIME, in *aiDM '19*, Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450368025. doi:10.1145/3329859.3329878.
[107] X. Mao, W. Wang, Y. Wu and M. Lan, LightEA: A Scalable, Robust, and Interpretable Entity Alignment Framework via Three-view Label Propagation, arXiv, 2022. doi:10.48550/ARXIV.2210.10436. https://arxiv.org/abs/2210.10436.
[108] Z. Yao, C. Li, T. Dong, X. Lv, J. Yu, L. Hou, J. Li, Y. Zhang and Z. Dai, Interpretable and Low-Resource Entity Matching via Decoupling Feature Learning from Decision Making, *CoRR* **abs/2106.04174** (2021). https://arxiv.org/abs/2106.04174.
[109] T. Deng, L. Hou and Z. Han, Keys as Features for Graph Entity Matching, in: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 1974–1977. doi:10.1109/ICDE48307.2020.00217.
[110] W. Zhang, S. Deng, H. Wang, Q. Chen, W. Zhang and H. Chen, XTransE: Explainable Knowledge Graph Embedding for Link Prediction with Lifestyles in e-Commerce, in: *Semantic Technology*, X. Wang, F.A. Lisi, G. Xiao and E. Botoeva, eds, Springer Singapore, Singapore, 2020, pp. 78–87. doi:10.1007/978-981-15-3412-6_8.
[111] J. Stadelmaier and S. Padó, Modeling Paths for Explainable Knowledge Base Completion, in: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 147–157. doi:10.18653/v1/W19-4816. https://aclanthology.org/W19-4816.
[112] W. Zhang, B. Paudel, W. Zhang, A. Bernstein and H. Chen, Interaction Embeddings for Prediction and Explanation in Knowledge Graphs, in *WSDM '19*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 96–104–. ISBN 9781450359405. doi:10.1145/3289600.3291014.
[113] U. Zulaika, A. Almeida and D. López-de-Ipiña, Influence Functions for Interpretable link prediction in Knowledge Graphs for Intelligent Environments, in: *2022 7th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2022, pp. 1–7. doi:10.23919/SpliTech55088.2022.9854264.
[114] W. Jiang, Y. Fu, H. Zhao, J. Wan and S. Pu, Graph Intention Neural Network for Knowledge Graph Reasoning, in: *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8. doi:10.1109/IJCNN55064.2022.9892730.
[115] M.K. Islam, S. Aridhi and M. Smail-Tabbone, Negative sampling and rule mining for explainable link prediction in knowledge graphs, *Knowledge-Based Systems* **250** (2022), 109083. doi:https://doi.org/10.1016/j.knosys.2022.109083. https://www.sciencedirect.com/science/article/pii/S0950705122005342.
[116] C. d'Amato, P. Masella and 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, pp. 170–177–. ISBN 9781450391153. doi:10.1145/3486622.3493956.
[117] B.Y. Lin, D. Lee, M. Shen, R. Moreno, X. Huang, P. Shiralkar and X. Ren, TriggerNER: Learning with Entity Triggers as Explanations for Named Entity Recognition, *CoRR* **abs/2004.07493** (2020). https://arxiv.org/abs/2004.07493.
[118] D. Lee, R.K. Selvam, S.M. Sarwar, B.Y. Lin, M. Agarwal, F. Morstatter, J. Pujara, E. Boschee, J. Allan and X. Ren, AutoTriggER: Named Entity Recognition with Auxiliary Trigger Extraction, *CoRR* **abs/2109.04726** (2021). https://arxiv.org/abs/2109.04726.
[119] H. Ouchi, J. Suzuki, S. Kobayashi, S. Yokoi, T. Kuribayashi, R. Konno and K. Inui, Instance-Based Learning of Span Representations: A Case Study through Named Entity Recognition, *CoRR* **abs/2004.14514** (2020). https://arxiv.org/abs/2004.14514.
[120] H. Shahbazi, X. Fern, R. Ghaeini and P. Tadepalli, Relation Extraction with Explanation, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, 2020, pp. 6488–6494. doi:10.18653/v1/2020.acl-main.579. https://aclanthology.org/2020.acl-main.579.
[121] A. Albalak, V. Embar, Y. Tuan, L. Getoor and W.Y. Wang, D-REX: Dialogue Relation Extraction with Explanations, *CoRR* **abs/2109.05126** (2021). https://arxiv.org/abs/2109.05126.
[122] S. Zeng, Y. Wu and B. Chang, SIRE: Separate Intra- and Inter-sentential Reasoning for Document-level Relation Extraction, *CoRR* **abs/2106.01709** (2021). https://arxiv.org/abs/2106.01709.
[123] Y. Xiao, Z. Zhang, Y. Mao, C. Yang and J. Han, SAIS: Supervising and Augmenting Intermediate Steps for Document-Level Relation Extraction, *CoRR* **abs/2109.12093** (2021). https://arxiv.org/abs/2109.12093.
[124] W. Zhou, H. Lin, B.Y. Lin, Z. Wang, J. Du, L. Neves and X. Ren, NERO: A Neural Rule Grounding Framework for Label-Efficient Relation Extraction, in: *Proceedings of The Web Conference 2020*, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2166–2176–. ISBN 9781450370233. doi:10.1145/3366423.3380282.
[125] H. Wang, K. Qin, G. Lu, J. Yin, R.Y. Zakari and J.W. Owusu, Document-level relation extraction using evidence reasoning on RST-GRAPH, *Knowledge-Based Systems* **228** (2021), 107274. doi:https://doi.org/10.1016/j.knosys.2021.107274. https://www.sciencedirect.com/science/article/pii/S0950705121005360.

[126] H. Kilicoglu, G. Rosemblat, M. Fiszman and D. Shin, Broad-coverage biomedical relation extraction with SemRep, *BMC Bioinformatics* **21**(1) (2020), 188. doi:10.1186/s12859-020-3517-7.

[127] D. Ru, C. Sun, J. Feng, L. Qiu, H. Zhou, W. Zhang, Y. Yu and L. Li, Learning Logic Rules for Document-level Relation Extraction, *CoRR* **abs/2111.05407** (2021). https://arxiv.org/abs/2111.05407.

[128] J. Yeo, H. Park, S. Lee, E.W. Lee and S.-w. Hwang, XINA: Explainable Instance Alignment Using Dominance Relationship, *IEEE Transactions on Knowledge and Data Engineering* **32**(2) (2020), 388–401. doi:10.1109/TKDE.2018.2881956.

[129] R. Singh, V.V. Meduri, A. Elmagarmid, S. Madden, P. Papotti, J.-A. Quiané-Ruiz, A. Solar-Lezama and N. Tang, Synthesizing Entity Matching Rules by Examples, *Proc. VLDB Endow.* **11**(2) (2017), 189–202–. doi:10.14778/3149193.3149199.

[130] D. Neil, J. Briody, A. Lacoste, A. Sim, P. Creed and A. Saffari, Interpretable Graph Convolutional Neural Networks for Inference on Noisy Knowledge Graphs, *CoRR* **abs/1812.00279** (2018). http://arxiv.org/abs/1812.00279.

[131] J. Jung, J. Jung and U. Kang, Learning to Walk across Time for Interpretable Temporal Knowledge Graph Completion, in: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 786–795–. ISBN 9781450383325. doi:10.1145/3447548.3467292.

[132] Y. Wang, H. Wang, J. He, W. Lu and S. Gao, TAGAT: Type-Aware Graph Attention neTworks for reasoning over knowledge graphs, *Knowledge-Based Systems* **233** (2021), 107500. doi:https://doi.org/10.1016/j.knosys.2021.107500. https://www.sciencedirect.com/science/article/pii/S0950705121007620.

[133] J. Wu, W. Shi, X. Cao, J. Chen, W. Lei, F. Zhang, W. Wu and X. He, DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network, in: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2140–2149–. ISBN 9781450384469. doi:10.1145/3459637.3482424.

[134] X. Yuan, Q. Lei, S. Yu, C. Xu and Z. Chen, Fine-Grained Relational Learning for Few-Shot Knowledge Graph Completion, *SIGAPP Appl. Comput. Rev.* **22**(3) (2022), 25–38–. doi:10.1145/3570733.3570735.

[135] J. Wu, S. Mai and H. Hu, Contextual relation embedding and interpretable triplet capsule for inductive relation prediction, *Neurocomputing* **505** (2022), 80–91. doi:https://doi.org/10.1016/j.neucom.2022.07.043. https://www.sciencedirect.com/science/article/pii/S0925231222008992.

[136] R. Bhowmik and G. de Melo, A Joint Framework for Inductive Representation Learning and Explainable Reasoning in Knowledge Graphs, *CoRR* **abs/2005.00637** (2020). https://arxiv.org/abs/2005.00637.

[137] W. Zhang, S. Deng, M. Chen, L. Wang, Q. Chen, F. Xiong, X. Liu and H. Chen, Knowledge Graph Embedding in E-Commerce Applications: Attentive Reasoning, Explanations, and Transferable Rules, in *IJCKG'21*, Association for Computing Machinery, New York, NY, USA, 2022, pp. 71–79–. ISBN 9781450395656. doi:10.1145/3502223.3502232.

[138] C. Meilicke, M.W. Chekol, M. Fink and H. Stuckenschmidt, Reinforced Anytime Bottom Up Rule Learning for Knowledge Graph Completion, *CoRR* **abs/2004.04412** (2020). https://arxiv.org/abs/2004.04412.

[139] Z. Han, P. Chen, Y. Ma and V. Tresp, xERTE: Explainable Reasoning on Temporal Knowledge Graphs for Forecasting Future Links, *CoRR* **abs/2012.15537** (2020). https://arxiv.org/abs/2012.15537.

[140] A. Sadeghian, M. Armandpour, P. Ding and D.Z. Wang, DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs, *CoRR* **abs/1911.00055** (2019). http://arxiv.org/abs/1911.00055.

[141] H. Sun, J. Zhong, Y. Ma, Z. Han and K. He, TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting, *CoRR* **abs/2109.04101** (2021). https://arxiv.org/abs/2109.04101.

[142] S. Ott, C. Meilicke and M. Samwald, SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models, *CoRR* **abs/2109.08002** (2021). https://arxiv.org/abs/2109.08002.

[143] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A.J. Smola and A. McCallum, Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning, *CoRR* **abs/1711.05851** (2017). http://arxiv.org/abs/1711.05851.

[144] Y. Liu, Y. Ma, M. Hildebrandt, M. Joblin and V. Tresp, TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs, *Proceedings of the AAAI Conference on Artificial Intelligence* **36**(4) (2022), 4120–4127. doi:10.1609/aaai.v36i4.20330. https://ojs.aaai.org/index.php/AAAI/article/view/20330.

[145] W. Xiong, T. Hoang and W.Y. Wang, DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning, *CoRR* **abs/1707.06690** (2017). http://arxiv.org/abs/1707.06690.

[146] P. Wang, K. Agarwal, C. Ham, S. Choudhury and C.K. Reddy, Self-Supervised Learning of Contextual Embeddings for Link Prediction in Heterogeneous Networks, in *WWW '21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2946–2957–. ISBN 9781450383127. doi:10.1145/3442381.3450060.

[147] H. Wang, H. Ren and J. Leskovec, Relational Message Passing for Knowledge Graph Completion, in: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1697–1707–. ISBN 9781450383325. doi:10.1145/3447548.3467247.

[148] Z. Du, C. Zhou, J. Yao, T. Tu, L. Cheng, H. Yang, J. Zhou and J. Tang, CogKR: Cognitive Graph for Multi-Hop Knowledge Reasoning, *IEEE Transactions on Knowledge and Data Engineering* **35**(2) (2023), 1283–1295. doi:10.1109/TKDE.2021.3104310.

[149] C. Lawrence, T. Sztyler and M. Niepert, Explaining Neural Matrix Factorization with Gradient Rollback, *CoRR* **abs/2010.05516** (2020). https://arxiv.org/abs/2010.05516.

[150] M. Qu, J. Chen, L.A.C. Xhonneux, Y. Bengio and J. Tang, RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs, *CoRR* **abs/2010.04029** (2020). https://arxiv.org/abs/2010.04029.

[151] C. Fu, T. Chen, M. Qu, W. Jin and X. Ren, Collaborative Policy Learning for Open Knowledge Graph Reasoning, *CoRR* **abs/1909.00230** (2019). http://arxiv.org/abs/1909.00230.

[152] Y. Gu, Y. Guan and P. Missier, Efficient Rule Learning with Template Saturation for Knowledge Graph Completion, *CoRR* **abs/2003.06071** (2020). https://arxiv.org/abs/2003.06071.

[153] G. Niu, B. Li, Y. Zhang and S. Pu, CAKE: A Scalable Commonsense-Aware Framework For Multi-View Knowledge Graph Completion, arXiv, 2022. doi:10.48550/ARXIV.2202.13785. https://arxiv.org/abs/2202.13785.

[154] M. Hildebrandt, J.A. Quintero Serna, Y. Ma, M. Ringsquandl, M. Joblin and V. Tresp, Reasoning on Knowledge Graphs with Debate Dynamics, *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(04) (2020), 4123–4131. doi:10.1609/aaai.v34i04.6600. https://ojs.aaai.org/index.php/AAAI/article/view/6600.

[155] Y. Zhang and Q. Yao, Knowledge Graph Reasoning with Relational Directed Graph, *CoRR* **abs/2108.06040** (2021). https://arxiv.org/abs/2108.06040.

[156] T. Ma, S. Lv, L. Huang and S. Hu, HiAM: A Hierarchical Attention based Model for knowledge graph multi-hop reasoning, *Neural Networks* **143** (2021), 261–270. doi:https://doi.org/10.1016/j.neunet.2021.06.008. https://www.sciencedirect.com/science/article/pii/S0893608021002409.

[157] Y. Xia, M. Lan, J. Luo, X. Chen and G. Zhou, Iterative rule-guided reasoning over sparse knowledge graphs with deep reinforcement learning, *Information Processing & Management* **59**(5) (2022), 103040. doi:https://doi.org/10.1016/j.ipm.2022.103040. https://www.sciencedirect.com/science/article/pii/S0306457322001492.

[158] G. Niu, B. Li, Y. Zhang, Y. Sheng, C. Shi, J. Li and S. Pu, Joint semantics and data-driven path representation for knowledge graph reasoning, *Neurocomputing* **483** (2022), 249–261. doi:https://doi.org/10.1016/j.neucom.2022.02.011. https://www.sciencedirect.com/science/article/pii/S0925231222001515.

[159] A. Zhu, D. Ouyang, S. Liang and J. Shao, Step by step: A hierarchical framework for multi-hop knowledge graph reasoning with reinforcement learning, *Knowledge-Based Systems* **248** (2022), 108843. doi:https://doi.org/10.1016/j.knosys.2022.108843. https://www.sciencedirect.com/science/article/pii/S0950705122004026.

[160] D. Lei, G. Jiang, X. Gu, K. Sun, Y. Mao and X. Ren, Learning Collaborative Agents with Rule Guidance for Knowledge Graph Reasoning, *CoRR* **abs/2005.00571** (2020). https://arxiv.org/abs/2005.00571.

[161] G. Niu, Y. Zhang, B. Li, P. Cui, S. Liu, J. Li and X. Zhang, Rule-Guided Compositional Representation Learning on Knowledge Graphs, *CoRR* **abs/1911.08935** (2019). http://arxiv.org/abs/1911.08935.

[162] C. Zhang, C.-N. Hsu, Y. Katsis, H.-C. Kim and Y. Vázquez-Baeza, Theoretical Rule-based Knowledge Graph Reasoning by Connectivity Dependency Discovery, in: *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–9. doi:10.1109/IJCNN55064.2022.9891938.

[163] P. Betz, C. Meilicke and H. Stuckenschmidt, Supervised Knowledge Aggregation for Knowledge Graph Completion, in: *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, P. Groth, M. Vidal, F.M. Suchanek, P.A. Szekely, P. Kapanipathi, C. Pesquita, H. Skaf-Molli and M. Tamper, eds, Lecture Notes in Computer Science, Vol. 13261, Springer, 2022, pp. 74–92. doi:10.1007/978-3-031-06981-9_5.

[164] T. Rocktäschel and S. Riedel, End-to-end Differentiable Proving, in: *Advances in Neural Information Processing Systems*, Vol. 30, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper/2017/file/b2ab001909a8a6f04b51920306046ce5-Paper.pdf.

[165] Y. Bai, X. Lv, J. Li, L. Hou, Y. Qu, Z. Dai and F. Xiong, SQUIRE: A Sequence-to-sequence Framework for Multi-hop Knowledge Graph Reasoning, *CoRR* **abs/2201.06206** (2022). https://arxiv.org/abs/2201.06206.

[166] G. Niu and B. Li, Logic and Commonsense-Guided Temporal Knowledge Graph Completion, arXiv, 2022. doi:10.48550/ARXIV.2211.16865. https://arxiv.org/abs/2211.16865.

[167] M.A. Hedderich, J. Fischer, D. Klakow and J. Vreeken, Label-Descriptive Patterns and their Application to Characterizing Classification Errors, *CoRR* **abs/2110.09599** (2021). https://arxiv.org/abs/2110.09599.

[168] A. Ebaid, S. Thirumuruganathan, W.G. Aref, A. Elmagarmid and M. Ouzzani, EXPLAINER: Entity Resolution Explanations, in: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019, pp. 2000–2003. doi:10.1109/ICDE.2019.00224.

[169] A. Zupon, M. Alexeeva, M. Valenzuela-Escárcega, A. Nagesh and M. Surdeanu, Lightly-supervised Representation Learning with Global Interpretability, in: *Proceedings of the Third Workshop on Structured Prediction for NLP*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 18–28. doi:10.18653/v1/W19-1504. https://aclanthology.org/W19-1504.

[170] N. Ding, X. Wang, Y. Fu, G. Xu, R. Wang, P. Xie, Y. Shen, F. Huang, H. Zheng and R. Zhang, Prototypical Representation Learning for Relation Extraction, *CoRR* **abs/2103.11647** (2021). https://arxiv.org/abs/2103.11647.

[171] D.J.T. Cucala, B.C. Grau, E.V. Kostylev and B. Motik, Explainable GNN-Based Models over Knowledge Graphs, in: *International Conference on Learning Representations*, 2022. https://openreview.net/forum?id=CrCvGNHAIrz.

[172] P. Pezeshkpour, Y. Tian and S. Singh, Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications, *CoRR* **abs/1905.00563** (2019). http://arxiv.org/abs/1905.00563.

[173] Q. Xie, X. Ma, Z. Dai and E.H. Hovy, An Interpretable Knowledge Transfer Model for Knowledge Base Completion, *CoRR* **abs/1704.05908** (2017). http://arxiv.org/abs/1704.05908.

[174] H. Lu, H. Hu and X. Lin, DensE: An enhanced non-commutative representation for knowledge graph embedding with adaptive semantic hierarchy, *Neurocomputing* **476** (2022), 115–125. doi:https://doi.org/10.1016/j.neucom.2021.12.079. https://www.sciencedirect.com/science/article/pii/S0925231221019342.

[175] A. Bastos, K. Singh, A. Nadgeri, S. Shekarpour, I.O. Mulang and J. Hoffart, HopfE: Knowledge Graph Representation Learning Using Inverse Hopf Fibrations, in: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 89–99–. ISBN 9781450384469. doi:10.1145/3459637.3482263.

[176] Y. Wang, H. Wang, W. Lu and Y. Yan, METransE: Manifold-like mechanism enhanced embedding for reasoning over knowledge graphs, *Expert Systems with Applications* **209** (2022), 118288. doi:https://doi.org/10.1016/j.eswa.2022.118288. https://www.sciencedirect.com/science/article/pii/S0957417422014245.

[177] S. Vadrevu, R. Nagi, J. Xiong and W.-m. Hwu, xER: An Explainable Model for Entity Resolution using an Efficient Solution for the Clique Partitioning Problem, in: *Proceedings of the First Workshop on Trustworthy Natural Language Processing*, Association for Computational Linguistics, Online, 2021, pp. 34–44. doi:10.18653/v1/2021.trustnlp-1.5. https://aclanthology.org/2021.trustnlp-1.5.

[178] Q. Lin, R. Mao, J. Liu, F. Xu and E. Cambria, Fusing topology contexts and logical rules in language models for knowledge graph completion, *Information Fusion* **90** (2023), 253–264. doi:https://doi.org/10.1016/j.inffus.2022.09.020. https://www.sciencedirect.com/science/article/pii/S1566253522001592.

[179] F. Yang, Z. Yang and W.W. Cohen, Differentiable Learning of Logical Rules for Knowledge Base Completion, *CoRR* **abs/1702.08367** (2017). http://arxiv.org/abs/1702.08367.

[180] A. Meroño-Pe nuela, R. Pernisch, C. Guéret and S. Schlobach, Multi-Domain and Explainable Prediction of Changes in Web Vocabularies, in: *Proceedings of the 11th on Knowledge Capture Conference*, K-CAP '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 193–200–. ISBN 9781450384575. doi:10.1145/3460210.3493583.

[181] T.-K. Tran, M.H. Gad-Elrab, D. Stepanova, E. Kharlamov and J. Strötgen, Fast Computation of Explanations for Inconsistency in Large-Scale Knowledge Graphs, in: *Proceedings of The Web Conference 2020*, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2613–2619–. ISBN 9781450370233. doi:10.1145/3366423.3380014.

[182] M. Kejriwal, R. Shao and P. Szekely, Expert-Guided Entity Extraction Using Expressive Rules, in *SIGIR'19*, Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450361729. doi:10.1145/3331184.3331392.

[183] K. Qian, L. Popa and P. Sen, SystemER: A Human-in-the-Loop System for Explainable Entity Resolution **12**(12) (2019), 1794–1797–. doi:10.14778/3352063.3352068.

[184] M. Paganelli, P. Sottovia, F. Guerra and Y. Velegrakis, TuneR: Fine Tuning of Rule-Based Entity Matchers, in: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2945–2948–. ISBN 9781450369763. doi:10.1145/3357384.3357854.

[185] Y. He, J. Chen, D. Antonyrajah and I. Horrocks, BERTMap: A BERT-based Ontology Alignment System, *CoRR* **abs/2112.02682** (2021). https://arxiv.org/abs/2112.02682.

[186] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention Is All You Need, *CoRR* **abs/1706.03762** (2017). http://arxiv.org/abs/1706.03762.

[187] M. Kejriwal, A meta-engine for building domain-specific search engines, *Software Impacts* **7** (2021), 100052. doi:https://doi.org/10.1016/j.simpa.2020.100052. https://www.sciencedirect.com/science/article/pii/S2665963820300439.

[188] G. Plumb, D. Molitor and A. Talwalkar, Model agnostic supervised local explanations, in: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 2520–2529–. http://arxiv.org/abs/1807.02910.

[189] D. Bahdanau, K. Cho and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015. http://arxiv.org/abs/1409.0473.

[190] R. Singh, V. Meduri, A. Elmagarmid, S. Madden, P. Papotti, J.-A. Quiané-Ruiz, A. Solar-Lezama and N. Tang, Generating Concise Entity Matching Rules, in: *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1635–1638–. ISBN 9781450341974. doi:10.1145/3035918.3058739.

[191] M. Ivanovs, R. Kadikis and K. Ozols, Perturbation-based methods for explaining deep neural networks: A survey, *Pattern Recognition Letters* **150** (2021), 228–234. doi:https://doi.org/10.1016/j.patrec.2021.06.030. https://www.sciencedirect.com/science/article/pii/S0167865521002440.

[192] V. Dignum, *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, 1st edn, Springer Publishing Company, Incorporated, 2019. ISBN 3030303705. doi:10.1007/978-3-030-30371-6.

[193] S. Hooker, D. Erhan, P.-J. Kindermans and B. Kim, A Benchmark for Interpretability Methods in Deep Neural Networks, in: *Advances in Neural Information Processing Systems*, Vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds, Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/fe4b8556000d0f0cae99daa5c5c5a410-Paper.pdf.

[194] X. Lv, Y. Cao, L. Hou, J. Li, Z. Liu, Y. Zhang and Z. Dai, Is Multi-Hop Reasoning Really Explainable? Towards Benchmarking Reasoning Interpretability, in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 8899–8911. doi:10.18653/v1/2021.emnlp-main.700. https://aclanthology.org/2021.emnlp-main.700.

[195] M.L. Leavitt and A.S. Morcos, Towards falsifiable interpretability research, *CoRR* **abs/2010.12016** (2020). https://arxiv.org/abs/2010.12016.

[196] M.T. Ribeiro, S. Singh and C. Guestrin, Anchors: High-Precision Model-Agnostic Explanations, *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1) (2018). doi:10.1609/aaai.v32i1.11491. https://ojs.aaai.org/index.php/AAAI/article/view/11491.

[197] B. Letham, C. Rudin, T.H. McCormick and D. Madigan, Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model, *The Annals of Applied Statistics* **9**(3) (2015), 1350–1371. doi:10.1214/15-AOAS848.

[198] P. Choudhary, A. Kramer and datascience.com team, datascienceinc/Skater: Enable Interpretability via Rule Extraction(BRL), Zenodo, 2018. doi:10.5281/zenodo.1198885.

[199] A. Crisan, M. Drouhard, J. Vig and N. Rajani, Interactive Model Cards: A Human-Centered Approach to Model Documentation, in: *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 427–439–. ISBN 9781450393522. doi:10.1145/3531146.3533108.

[200] L. Asprino, E. Daga, A. Gangemi and P. Mulholland, Knowledge Graph Construction with a Façade: A Unified Method to Access Heterogeneous Data Sources on the Web, *ACM Trans. Internet Technol.* (2022). doi:10.1145/3555312.

[201] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E.H. Chi, Q. Le and D. Zhou, Chain of Thought Prompting Elicits Reasoning in Large Language Models, *CoRR* **abs/2201.11903** (2022). https://arxiv.org/abs/2201.11903.

[202] S. Hellmann, J. Lehmann, S. Auer and M. Brümmer, Integrating NLP Using Linked Data, in: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J.X. Parreira, L. Aroyo, N.F. Noy, C. Welty and K. Janowicz, eds, Lecture Notes in Computer Science, Vol. 8219, Springer, 2013, pp. 98–113. doi:10.1007/978-3-642-41338-4_7.

[203] C. Di Bonaventura, L. Siciliani, P. Basile, A. Merono Penuela and B. McGillivray, *Is Explanation All You Need? An Expert Survey on LLM-generated Explanations for Abusive Language Detection*, in: *Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024. https://ceur-ws.org/Vol-3878/33_main_long.pdf.

[204] E.F. Kendall and D.L. McGuinness, Ontology engineering, *Synthesis Lectures on the Semantic Web: Theory and Technology* **9**(1) (2019), i–102. doi:10.1007/978-3-031-79486-5.

[205] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, The NeOn methodology for ontology engineering, in: *Ontology engineering in a networked world*, Springer, 2011, pp. 9–34. doi:10.1007/978-3-642-24794-1_2.

[206] A. Halfaker and R.S. Geiger, ORES: Lowering Barriers with Participatory Machine Learning in Wikipedia, *CoRR* **abs/1909.05189** (2019). http://arxiv.org/abs/1909.05189.