Information Processing and Management 62 (2025) 104145

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Knowledge graph validation by integrating LLMs and human-in-the-loop

Stefani Tsaneva [a,*], Danilo Dessì [b,1], Francesco Osborne [c,d,1], Marta Sabou [a]

[a] Institute of Data, Process and Knowledge Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020, Vienna, Austria
[b] Computer Science Department, College of Computing and Informatics, University of Sharjah, University City, Sharjah, United Arab Emirates
[c] Knowledge Media Institute, The Open University, Walton Hall, MK7 6AA, Milton Keynes, United Kingdom
[d] Department of Business and Law, University of Milano Bicocca, Via Bicocca degli Arcimboldi 8, 20126, Milan, Italy

ARTICLE INFO

Dataset link: https://github.com/danilo-dessi/SKG-pipeline/blob/main/eval/, https://doi.org/10.5281/zenodo.13730203

Keywords:
Knowledge graph validation
Large language models
Hybrid human-AI workflows

ABSTRACT

Ensuring the quality of knowledge graphs (KGs) is crucial for the success of the intelligent applications they support. Recent advances in large language models (LLMs) have demonstrated human-level performance across various tasks, raising the question of their potential for KG validation. In this work, we explore the role of LLMs in human-centric KG validation workflows, examining different collaboration strategies between LLMs and domain experts. We propose and evaluate nine distinct approaches, ranging from fully automated validation to hybrid methods that combine expert oversight with AI assistance. These workflows are tested within a real-world KG construction pipeline used to generate the Computer Science Knowledge Graph (CS-KG), a large-scale resource designed to support scientometric tasks such as trend forecasting and hypothesis generation. CS-KG comprises 41 million statements represented as 350 million triples within the Computer Science domain. Our findings show that integrating LLMs into the CS-KG verification process enhances precision by 12%, improving alignment with expert-level validation. However, this comes at the cost of recall, resulting in a 5% decrease in the overall F1 score. In contrast, a hybrid approach which involves both human-in-the-loop and LLM modules, yields the best overall results, improving F1 score by 5% with minimal human involvement.

1. Introduction

Knowledge graphs (KGs) are conceptual models that structure domain knowledge, integrated from various sources, and stored in a machine-readable and understandable format (Hogan et al., 2021; Peng, Xia, Naseriparsa, & Osborne, 2023). KGs are employed in a variety of intelligent applications (Paulheim, 2017) supporting tasks such as question answering (Yani & Krisnadhi, 2021), recommender systems (Guo et al., 2020), and exploratory search (Nuzzolese, Presutti, Gangemi, Peroni, & Ciancarini, 2017). KG-based solutions have been adopted across various domain such as medicine (Li et al., 2020), production & manufacturing (Wang, Cheng, Qi, & Tao, 2024; Xu & Dang, 2023), tourism (Chessa et al., 2023), and education (Su & Zhang, 2020). In the scientometrics domain, scientific KGs have recently gained significant interest as a solution for knowledge-based content exploration of scientific works (Dessí, Osborne, Reforgiato Recupero, Buscaldi, & Motta, 2022b; Dessì et al., 2020; Jaradeh et al., 2019; Meloni et al., 2023;

* Corresponding author.
E-mail addresses: stefani.tsaneva@wu.ac.at (S. Tsaneva), danilo.dessi@gesis.org (D. Dessì), francesco.osborne@open.ac.uk (F. Osborne), marta.sabou@wu.ac.at (M. Sabou).
[1] These authors contributed equally.

https://doi.org/10.1016/j.ipm.2025.104145
Received 14 September 2024; Received in revised form 26 February 2025; Accepted 10 March 2025
Available online 9 April 2025

Sakor et al., 2023). Some of the proposed scientific knowledge graphs have been manually curated, ensuring the high quality of these resources, such as the Open Research Knowledge Graph (Jaradeh et al., 2019). Others prioritize a high coverage of the scientific domain and have been generated through automated approaches, e.g., COVID-19 Knowledge Graph (Sakor et al., 2023) and the Computer Science Knowledge Graph (Dessí et al., 2022b).

While automated KG generation allows the integration of content from a vast amount of sources and provides extensive coverage of a given domain, the produced resources can face quality issues such as the inclusion of erroneous, inconsistent or misleading facts (Paulheim, 2017; Zaveri et al., 2013). To ensure the success of systems relying on the constructed KGs, knowledge graph validation is an essential step to be integrated within KG generation pipelines.

Against this backdrop, various approaches for KG quality evaluation have been proposed. In Xue and Zou (2023), the authors provide an overview of state-of-the-art validation techniques, categorized as automated methods relying on statistics (e.g., outlier detection and KG embedding) or rules (e.g., ontology-based rules), and methods relying on a human-in-the-loop (HiL), such as crowdsourcing. While some KG generation pipelines contain an integrated automated validation stage, HiL-based KG validation techniques do not scale well (Paulheim, 2017; Xue & Zou, 2023) and are thus often excluded or envisioned as a future extension.

Recently, large language models (LLMs) have demonstrated human-like performance in a variety of natural language processing tasks, significantly reducing the need of human intervention (Chiang & Lee, 2023; Sallam, Al-Salahat, Eid, Egger, & Puladi, 2024). As a result, several works in the knowledge engineering domain, e.g., Allen and Groth (2024), Fathallah et al. (2024), Khorashadizadeh, Mihindukulasooriya, Tiwari, Groppe, and Groppe (2023), Tsaneva, Vasic, and Sabou (2024), have been inspired, reporting promising LLM performance in the evaluation of semantic resources (i.e., ontologies, KGs). However, the conducted experiments are limited in terms of the simplicity of the application domain, small size of the evaluated resource and lack of comprehensive experimental investigations. Moreover, many open questions remain: (i) how should the proposed LLM-based approaches be evaluated? (ii) how should LLM-based approaches be integrated into the generation of real KGs? (iii) could these solutions fully replace or only support human validators?

This paper primarily addresses question (iii) above, exploring how to best combine LLMs and HiL for KG validation. Specifically, we investigate the following two research questions.

RQ1: What are different ways to combine LLMs and HiL contributors for the validation of large knowledge graphs? We investigate novel KG validation workflows, incorporating both HiL and LLM validations, to improve the performance of KG generation pipelines. We base our work on previous research on collaborative human-LLM workflows from a spectrum of automation levels, proposed for relevance judgments tasks in Faggioli et al. (2023), which we adopt for KG validation. As a result, we propose *nine distinct validation workflows*- three workflows that rely exclusively on human judgment, three hybrid solutions involving a combination of human expertise and LLMs, and three fully automated LLM-based validation pipelines. Each workflow is tailored towards a specific evaluation goal and availability of HiL and LLM resources to provide adaptability across different use cases.

RQ2: What are the strengths and limitations of human-LLM collaborative knowledge graph validation workflows? To the best of our knowledge, no prior empirical investigation has been performed of the trade-offs involved in hybrid human-machine workflows for KG validation. Thus, we conduct *two experimental investigations on a large scale resource*, representing a non-trivial domain, to collect empirical evidence and facilitate a direct comparison of the achieved performance with each workflow.

To explore RQ1 and RQ2, we consider the validation of the Computer Science Knowledge Graph (CS-KG) as our use case. CS-KG is a scientific knowledge graph, automatically generated from 6.7M publications, supporting researchers and funding agencies by enabling the exploration of research dynamics (Dessí et al., 2022b). We selected this KG for its relevance to various applications, its broad coverage of scientific concepts and domains, and its focus on a field well-known to the authors. Furthermore, CS-KG was constructed using an open methodology that already integrates a few validation techniques, which we can leverage in our analysis. Specifically, CS-KG was generated using the SCICERO pipeline (Dessí, Osborne, Recupero, Buscaldi, & Motta, 2022a), which extracts scientific statements from literature and represents them as triples of the form <*subject, predicate, object*>, for instance, <*cloud service, acquires, information integration*> or <*text classification, includes, text processing*>. SCICERO includes an automated validation stage, which we extend by integrating it with HiL techniques, LLM-sourced validations or a combination thereof. Subsequently we evaluate the achieved performance of these SCICERO extensions on a set of $3.6K$ triples.

Our results indicate that: (1) an LLM-based validation can increase precision from 75% up to 87% without requiring any manual validation efforts; (2) both fully manual and fully automated validation approaches present trade-offs between precision and recall; and (3) a hybrid approach, leveraging a HiL only upon a disagreement among automated methods, leads to smaller precision improvement up to 80% and overall highest F1 score reaching 82% (+5% compared to SCICERO) with minimal manual efforts.

The remainder of this paper is structured as follows. Section 2 reviews related research in the area. Section 3 introduces CS-KG and its extraction pipeline SCICERO. In Section 4, we propose extensions of SCICERO, covering a spectrum of automation levels from purely HiL-based validation to purely LLM-based validation. Section 5 details the design of two experiments carried out to evaluate these extended workflows. The experiment results are discussed in Section 6, followed by a conclusion and future work directions in Section 7.

2. Related work

While human-LLM frameworks have not yet been proposed for the validation of knowledge graphs, some semi-automatic approaches towards a scalable validation of semantic resources (i.e., ontologies, knowledge graphs) have been designed. Section 2.1 reviews several such works, providing an overview of state-of-the-art human-in-the-loop approaches. In Section 2.2, we explore automatic triple validation methods, with a particular focus on recent LLM-based techniques designed to enhance the quality of semantic resources. Finally, Section 2.3 lays the groundwork for the human-LLM workflows proposed in this paper by providing an overview of studies that examine the levels of collaboration in human-machine workflows across various domains.

2.1. Semantic resources evaluation workflows

A fully manual KG creation process, which involves trained domain experts and knowledge engineers, can produce higher quality resources than a fully automated approach. However, scalability becomes a significant challenge, especially for large-scale KGs. As an alternative, a human-centric KG validation, typically implying the annotation of triples as true or false by human contributors, can be included in an automated creation workflow to ensure the removal of incorrectly represented statements. Several research directions have emerged, trying to approach this issue from different angles.

Semi-automatic KG generation. Human-in-the-loop approaches have been incorporated as a final step in KG extraction workflows containing some level of automation. For instance, in Lossio-Ventura et al. (2018), HiL validation is carried out as part of the triple extraction step from medical literature to eliminate noisy triples. A similar workflow is also described in Rumin1 and Mekterović (2019), where human judgment is added as the last stage of the KG extraction workflow. While these works avoid the manual efforts of creating the KG, they introduce a bottleneck at the validation stage.

Triple selection for HiL annotation. Several approaches have been implemented to reduce the amount of triples requiring manual validation. In Demartini, Difallah, and Cudré-Mauroux (2013), HiL annotation is conducted to verify the results of an entity-linking prediction task, with triples being validated only if the prediction confidence score falls below a certain threshold. A similar approach combined with further contradiction reasoning is employed in Li et al. (2017) to select which triples to manually annotate.

Triple prioritization for HiL annotation. The minimization of triples to be checked by a HiL is also discussed in Ojha and Talukdar (2017), where triples are prioritized based on the amount of additional triples, whose correctness can be inferred from the annotation. This line of work is continued in Gao et al. (2019) and Qi, Zheng, Hong, and Zou (2022) by optimizing the cost and duration of the manual annotations and computational efforts.

HiL assistance. A semi-automatic workflow focusing on assisting human validators is presented in Pomp, Lipp, and Meisen (2019). In this approach, human contributors are assisted by an automatic tool using reasoning to provide suggestions based on previously validated constraints and identified inconsistencies.

In the mentioned examples, the KG validation task is performed by the human annotators while automated approaches either generate the triples to be checked (Lossio-Ventura et al., 2018; Rumin1 & Mekterović, 2019), aim to reduce the amount of triples to be verified (Demartini et al., 2013; Gao et al., 2019; Li et al., 2017; Ojha & Talukdar, 2017; Qi et al., 2022), or assist the human annotator with automated suggestions (Pomp et al., 2019). In parallel, various automated KG validation techniques have been investigated, which we summarize next.

2.2. Automated validation of semantic resources

Automatic KG extraction approaches typically face a trade off between the scope and quality of the resulting resource (Paulheim, 2017). As a result, several research directions have emerged focusing on automated validation of semantic resources, specifically addressing tasks such as completion and error-detection (Paulheim, 2017).

Related work has approached these tasks following various methods such as utilizing KG embeddings (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013; Dettmers, Minervini, Stenetorp, & Riedel, 2018; Nickel, Tresp, & Kriegel, 2012), graph features (Borrego, Ayala, Hernández, Rivero, & Ruiz, 2021) or transformers (Dessí et al., 2022a; Jaradeh, Singh, Stocker, & Auer, 2021; Yao, Mao, & Luo, 2019), developed to classify newly generated triples as valid or invalid. For KG completion, for instance, in Jaradeh et al. (2021), a classifier is trained on existing KG triples and subsequently applied to assess the validity of newly generated triples. Similarly, in Dessí et al. (2022a), a classifier is trained on a reliable subset of the KG and then used to determine whether uncertain triples are erroneous.

Recently, however, LLMs have demonstrated impressive human-level performance on tasks, typically completed by human contributors across various domains (Chiang & Lee, 2023; Sallam et al., 2024) without additional training required. Therefore, next, we discuss relevant LLM-based approaches in the Semantic Web domain, with focus on the validation of semantic resources.

Validating semantic resources with LLMs. Over the last years LLMs have attracted much research interest from the Semantic Web community. LLMs have been widely explored for some knowledge engineering tasks such as the creation or completion of semantic resources, e.g., Carta et al. (2023), Trajanoska, Stojanov, and Trajanov (2023), Zhang, Reklos, Jain, Peñuela, and Simperl (2023), Zhu et al. (2024). However, the evaluation of semantic resources with LLMs has only recently received interest.

In Khorashadizadeh et al. (2023), the authors include a triple validation step, using LLMs, in their KG generation workflow. Yet, they do not provide quantitative details on the performance of this validation approach or compare it to other methods.

In Fathallah et al. (2024), the focus is on designing a prompt-chain for generating ontologies, including a validation stage where errors are identified through external services (such as OOPS Poveda-Villalón, Gómez-Pérez, & Suárez-Figueroa, 2014 and an ontology reasoner) and corrected by LLMs. The paper introduces several successfully corrected examples from the generation of the Wine Ontology.[2] However, the authors do not specify whether the validation performance was tested in a concrete experimental setup.

[2] Wine Ontology - https://www.w3.org/TR/owl-guide/wine.rdf

The identification of ontology modeling defects through LLMs has been investigated in Tsaneva, Vasic et al. (2024). While the study reports validation accuracy of 96%, the carried out experiment relied on a small dataset from the Pizza Ontology.[3]

In Tsaneva, Herwanto, and Sabou (2024), the authors propose the development of knowledge engineering task-specific assessment tests and evaluate various LLMs based on their ability to validate ontology axioms. The benchmark aims to assess LLM capabilities, drawing inspiration from qualification tests typically used for crowdworkers.

In Allen and Groth (2024), the authors investigate incorrect & missing class membership relations in ontologies using LLMs. They experiment with relations extracted from public knowledge graphs in the general domain and across various LLMs. However, the dataset size and examined relation types are limited.

Recently, in Regino and Dos Reis (2025), the LLM-based validation of newly generated KG triples with focus on inconsistency detection has been explored. The authors investigate four fundamental aspects: aligning classes and properties, standardizing URIs, ensuring semantic consistency, and verifying syntactic accuracy.

While the reviewed literature indicates the potential of LLMs for validating semantic resources, proposed approaches are still in preliminary stages, tested on a small dataset or lack experimental evaluation. Additionally, there is a lack of studies exploring how the designed solutions should be best integrated into existing KG generation pipelines, i.e., whether they could fully replace previous automated/manual approaches or they should serve as a complementary tool.

2.3. Human–machine collaboration workflows

In the knowledge graph validation field, a study classifies validation approaches into methods based on human-annotation, statistics/learning, rules, and hybrid approaches that combine two or more of these methods (Xue & Zou, 2023). The authors argue that hybrid approaches have the potential to overcome the limitations of each separate method. While human-machine collaboration lacks thorough investigations in the Semantic Web community, possible interactions between human contributors and automated approaches have received research interests in various communities.

Collaborative workflows, illustrating the roles and responsibilities in hybrid human-AI teams have been explored for moral decision making in the medical domain (van Stijn, Neerincx, ten Teije, & Vethman, 2021). The work highlights levels of involvement of each agent (human or AI) and the expected advantages and disadvantages in terms of workload, accountability, and ethical concerns.

A recent positioning paper identifies different levels of collaborations between human annotators and LLMs for relevance judgment tasks (Faggioli et al., 2023). The authors discuss the implications of using LLM annotations compared to a traditional HiL approach, considering factors such as budget and quality. They discuss potential benefits and scenario implementations across a spectrum from fully manual to fully automated approaches (Faggioli et al., 2023), organized in the following categories:

- *Human judgment* implies that (1) the annotations are completed manually by human participants, who perform the tasks without any support or (2) human annotators are supported by tools (e.g., document clustering) but ultimately remain the sole judges of the triple correctness.
- *AI assistance* can be implemented in various ways with different levels of responsibility. For instance, LLMs can be employed to generate summaries or other contextual information to help the human judges in their annotations. Moreover, a task partitioning can be established where each agent focuses on tasks suited for their capabilities.
- *Human verification* describes a human-in-the-loop workflow, where human participants judge the results of an automated approach and correct them if needed. A novel implementation, motivated by the "preference-testing" concept, suggests that two LLMs can provide judgments and a human participant can choose the more relevant example.
- *Fully automated* workflows treat LLM judgments as reliable sources, which can completely replace human judges.

In this study, we consider interaction workflows identified in the literature, and provide experimental results for various collaboration workflows applied to a concrete use case: the validation of automatically extracted triples, part of the Computer Science Knowledge Graph, which we describe in the next section.

3. Use case: Validating the computer science knowledge graph

The Computer Science Knowledge Graph (CS-KG) describes a vast collection of claims extracted from 6.7 million scientific articles in the field of Computer Science. In CS-KG, scientific claims are represented as triples in the form *<subject, predicate, object>*, describing the relation (predicate) between two entities (subject and object). The knowledge graph contains about 10M entities, classified within the five categories *Method, Task, Material, Metric,* and *OtherEntity*, and connected through 179 object properties (Dessí et al., 2022a, 2022b).

For instance, the triple *<support vector machine,outperforms,decision tree>,* refers to the claim that the entity *support vector machine* of type *Method* outperforms the entity *decision tree* of type *Method*. Since often no objective truth can be determined, as in the given example, CS-KG claims should be considered only within the context of the articles, they are linked to Dessí et al. (2022b).

[3] Pizza Ontology - https://protege.stanford.edu/ontologies/pizza/pizza.owl

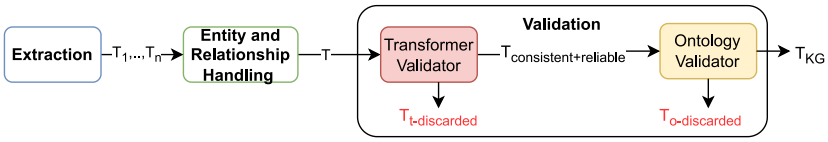

Fig. 1. The SCICERO pipeline, used to generate the Computer Science Knowledge Graph.

CS-KG supports a variety of tasks such as advanced literature search and trend forecasting by integrating content from various sources. Thus, the coverage of the knowledge graph is of high importance. However, it is equally important to identify and discard erroneous facts before they are incorporated into the final version of the knowledge graph.

In this section, we first explain the rationale for selecting CS-KG as our use case. Next, we present SCICERO, the generation pipeline employed to construct CS-KG. Finally, we describe the evaluation process used to assess the quality of the resulting resource.

Use case motivation. We selected CS-KG as our use case for two main reasons:

- *Significance and Adoption.* CS-KG has attracted significant research interest and has been integrated into various applications, demonstrating its relevance and value as a resource. For instance, it has been incorporated into a research support system proposed in Salatino, Mannocci, Osborne, Rehm, Schimmler, et al. (2024), which enables scientists to explore scientific literature, state-of-the-art methodologies for specific tasks, and available scientific artifacts. In Alonso, Dessí, Meloni, and Reforgiato Recupero (2025), the authors introduce a resume-to-job matching method that identifies the most suitable open positions based on an applicant's skills. They further propose an extension of their framework which leverages CS-KG to recommend relevant patents or articles, helping job candidates enhance their expertise in preparation for applications. CS-KG has also been employed in experimental investigations to facilitate the retrieval of papers within specific subfields of Computer Science (Kovacevic, El-Ebshihy, Piroi, & Rauber, 2024). Further research leveraging CS-KG and its predecessor, AI-KG (Dessì et al., 2020), includes entity extraction from scientific publications (Li & Daoutis, 2021), scholarly article classification (Hoppe, Dessì, & Sack, 2021), and knowledge graph completion (Borrego et al., 2022).
- *Ease of use for HiL evaluation.* The SCICERO pipeline was evaluated on a CS-KG subset containing $3.6K$ triples. This dataset includes not only the final ground truth label of each evaluated triple but also individual annotations from three experts. Consequently, the SCICERO gold standard enables the simulation of HiL experiments without requiring additional manual effort. Furthermore, the domain of CS-KG aligns with the expertise of the authors of this paper, facilitating access to a pool of knowledgeable experts who can support HiL approaches. These factors were instrumental in conducting the experiments presented in this study.

Generation of CS-KG. Fig. 1 displays the architecture of the SCICERO pipeline (Dessí et al., 2022a), which extracts triples from scientific literature to generate CS-KG. The pipeline takes as input a set of scientific texts related to the field of Computer Science, along with an ontology that defines the domain's semantics. SCICERO then generates knowledge graph triples through three main stages: extraction, entity and relationship handling, and validation.

In the *extraction* stage the framework applies the CSO classifier (Salatino, Osborne, & Motta, 2022), which identifies research topics described in a scientific publication, according to the Computer Science Ontology,[4] and revised NLP modules from the CoreNLP suite (Manning et al., 2014) to produce sets of initially extracted triples $T_1 ... T_n$. Extracted entities and relationships are further processed in *the entity and relationship handling* stage by, e.g., merging similar entities and discarding generic terms, thus resulting in an integrated set of triples, T. Lastly, the triple set is sent for a *validation* aiming to reduce noisy and erroneous triples. The validation stage contains two modules - a transformer-based validator and an ontology-based validator, which we will briefly describe next.

The *transformer validator* relies on the support of a triple, i.e., the number of textual sources from which the triple was extracted. As such the support-level s can be interpreted as confidence of the correctness of the triple. Following this intuition, the triple set T is split into $T_{reliable}$, containing triples with a high support (e.g., $s \geq 5$, that is triples which were extracted from 5 or more documents), and $T_{uncertain}$ (e.g., $s \leq 5$). Next, $T_{negative}$ is produced by corrupting triples of $T_{reliable}$ and a transformer model is fine-tuned with $T_{reliable}$ and $T_{negative}$. Lastly, a prediction is made for each triple from $T_{uncertain}$ resulting in two new sets of triples predicted as correct $T_{consistent}$ and incorrect $T_{t-discarded}$.

The *ontology validator* takes as input $T_{consistent}$ and $T_{reliable}$ and ensures the removal of triples which are not aligned with the domain ontology. Thus, a triple containing a subject or object of a type not specified as the domain and range for the triple relation are discarded ($T_{o-discarded}$). For instance, the ontology defines the relation *uses*, indicating that an entity instance of type *Method, Metric, OtherEntity, or Task* uses an instance of type *Method, Metric, OtherEntity, Task, or Material*.

The triple <*dbpedia, uses, core_nlp*> describes the relation between the entity *dbpedia*, classified as *Material*, and the entity *core_nlp* of type *Method*. The given triple does not comply to the ontological schema since no entity of type *Material* can use an entity of type *Method*. Thus, it will be discarded.

[4] Computer Science Ontology - https://scholkg.kmi.open.ac.uk/cskg/ontology

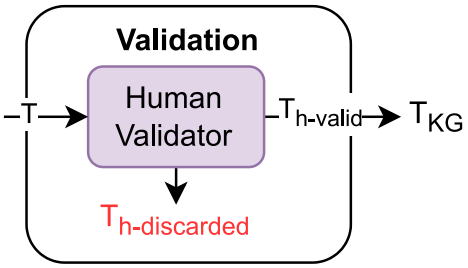

Fig. 2. Workflow 1: Human validation in place of the original SCICERO validation.

Evaluation. SCICERO and all its components have been evaluated on a subset of $3.6K$ triples. The KG generation achieves a precision of 75%, a recall of 79%, and an F1 score of 77%. The validation modules contribute to an over 20% increase in precision, highlighting the critical role of the validation stage. Additional information on the implementation of SCICERO and CS-KG is available in Dessí et al. (2022a) and Dessí et al. (2022b).

In this paper, we aim to further enhance the quality of the generated triples by looking into possible extensions of the SCICERO validation stage with new validation modules based on LLMs and human expertise.

4. Integrating LLMs and HiL into the SCICERO validation

Human validation has frequently been envisioned as an addition to automated KG generation workflows to ensure the quality of the produced knowledge graphs. We investigate possible extensions of the SCICERO pipeline with an additional *human validator* module. Building on recent advances in LLMs research, which have shown expert-level results for KG validation (Tsaneva, Vasic et al., 2024), we also investigate an additional *LLM validator* module. We explore both workflows, where the LLM validator fully replaces the human validator as well as scenarios where LLMs and human annotators collaborate in the KG validation task.

This section discusses potential SCICERO extension workflows, with a particular focus on the level of collaboration between distinct validation modules. Formally, let T be a set of input triples, S be the SCICERO pipeline, and V the outcome of the triple validation. We want to introduce one or both modules M_j, where $j \in \{HiL, LLM\}$, such that the enhanced pipeline $S+M_j$ archives better performance in terms of precision, recall, and F1 score, compared to S alone in validating the triples to be added to CS-KG. We build on top of our recent work (Tsaneva, Dessì, Osborne, & Sabou, 2024), where we perform a preliminary investigation of possible SCICERO extensions to select the most meaningful workflows which lead to the best performance scores. In the following section (Section 5) we discuss two experiments with concrete implementation details for the LLM and human validator modules.

In Table 1, we categorize potential extension workflows based on the level of collaboration between the integrated LLM and the human validator. Specifically, we classify these workflows according to the four categories introduced in Faggioli et al. (2023): human judgment, AI assistance, human verification, and full automation. Sections 4.1–4.4 provide a detailed description of each workflow and define the expected roles of the corresponding validation modules.

Additionally, we differentiate the types of the incorporated validation methods according to the classification from Xue and Zou (2023). In Figs. 2–10, statistics and learning-based methods are displayed in red. Rule-based methods are shown in yellow and human-based methods are displayed in purple.

4.1. Human judgment

We first explore possible positions of the human validator module within the SCICERO pipeline without involving an LLM. Since the original SCICERO pipeline already contains two automated validators, we consider several alternative configurations:

Human validation (workflow 1 in Table 1). For a completely manual annotation process without any tool-support the SCICERO pipeline needs to be modified. This involves replacing the entire SCICERO validation stage with a human validator module as illustrated in Fig. 2. While this workflow ensures full decision control by the human annotators, it is not a scalable solution for the validation of large resources such as CS-KG.

Human validation after SCICERO (workflow 2 in Table 1). An intuitive extension of SCICERO with human validation implies the positioning of the human validator module at the end of the SCICERO pipeline (see Fig. 3). In this scenario, the automated transformer and ontology validators can be seen as filters that the human judges utilize in order to reduce the pool of triples that need to be verified. The ontology validator relies on expert-defined rules that apply to the domain, while the transformer validator can be adjusted with the desired triple support level. By placing the human validator at the end, humans retain full control of what is included in the final KG while automated tools support the removal of noisy triples. This workflow is particularly useful when the primary validation goal is to further enhance the precision of the resulting KG triples.

Table 1
Possible extensions of SCICERO, employing various levels of interaction among human and LLM validations, following the interaction classification from Faggioli et al. (2023).

Collaboration level according to Faggioli et al. (2023)	Workflow ID	Figure	Workflow description
Human Judgment			
	1	Fig. 2	Human validation with no automated support.
	2	Fig. 3	Human validation as a final step of SCICERO's validation. Experts are supported in the removal of noisy triples but have full decision control over triple additions to the KG.
	3	Fig. 4	Partial human validation for low-support triples ($T_{consistent}$) as a final step of SCICERO's validation. Highly supported triples ($T_{consistent}$) are directly added to the KG.
AI Assistance			
	4	Fig. 5	Balanced competence partitioning. Human validation for low-support triples ($T_{consistent}$) and LLM validation of highly supported triples ($T_{consistent}$).
Human Verification			
	5	Fig. 6	Human validation triple correctness upon disagreement. LLM validation of $T_{t-discarded}$ and $T_{consistent}$ triples. Human validation whenever the LLM module disagrees with the original SCICERO validators.
	6	Fig. 7	Human validation upon triple removal disagreement. LLM validation of $T_{t-discarded}$. Human validation whenever the LLM disagrees with the original SCICERO validators.
Fully Automated			
	7	Fig. 8	LLM-based triples verification added to SCICERO's validation stage. LLMs take the final judgment.
	8	Fig. 9	LLM-based triples verification after SCICERO's original automated validation for $T_{consistent}$.
	9	Fig. 10	LLM-based triples verification. Replacement of the SCICERO validation through an LLM validation.

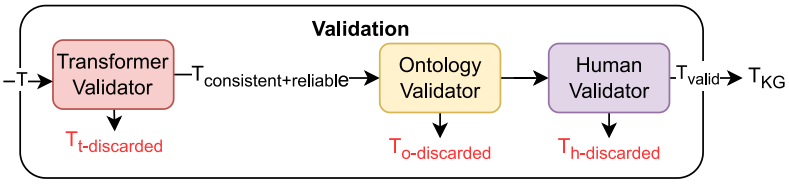

Fig. 3. Workflow 2: Human validation integration as a last step of the SCICERO validation.

Partial human validation after SCICERO (workflow 3 in Table 1). While sending all triples to the human validator module ensures the highest level of human oversight, this solution does not scale well for large resources since the manual efforts are enormous. Thus, we also consider a version of this workflow, where the human validator is only involved for the annotation of selected triples. Specifically, human annotation is added only for triples with limited literature support (Fig. 4). Although not every triple is manually reviewed before it is added to the KG, human oversight can be ensured by the selection of an appropriate reliability threshold for the transformer validator and establishing rules, to be followed by the ontology validator, with domain experts.

Following the task-partitioning idea from Faggioli et al. (2023), workflow 3 can also be classified as an example of *AI assistance* since human judges take control over $T_{consistent}$, while the automated validators takes care of $T_{reliable}$. However, in this paper we focus on the interaction between the human validator and LLM validator modules and thus consider the interaction in the workflow as *human judgment*. Next, we propose an extension of workflow 3 more closely fitting the AI (LLM) assistance paradigm.

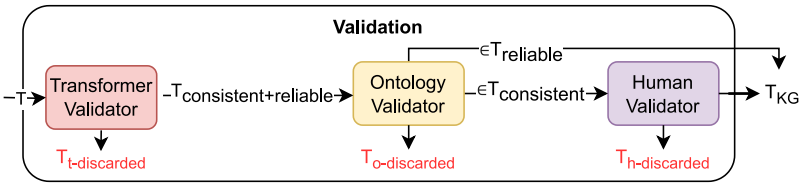

Fig. 4. Workflow 3: Partial human validation integration as a last step of the SCICERO validation for low-support triples ($T_{consistent}$).

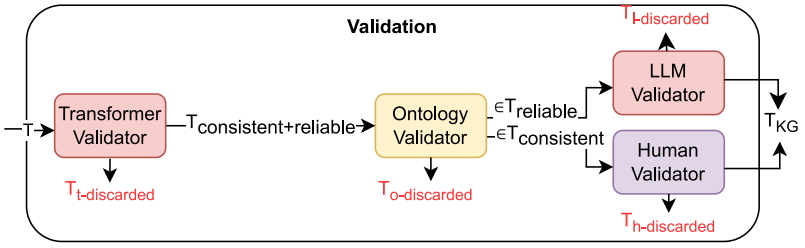

Fig. 5. Workflow 4: Balanced competence partitioning. Human judges validate $T_{consistent}$ triples while LLMs judge triples belonging to $T_{reliable}$.

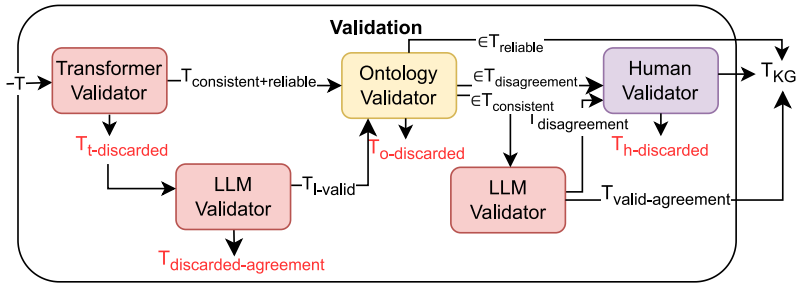

Fig. 6. Workflow 5: Human validation upon triple correctness disagreement among the LLM validator and original SCICERO validators.

4.2. AI assistance

Balanced competence partitioning (workflow 4 in Table 1). Intuitively, LLMs are likely to be more capable at validating statements that often occur in literature because of the availability of a larger training dataset. In contrast, human judges might struggle reviewing statement extracted from numerous sources, especially if contradictory results are presented. Therefore, we propose a workflow where LLMs deal with triples linked to a higher amount of scientific texts (i.e., triples with high support), while human participants focus on triples for which only a few references are available (i.e., triples with low support). Fig. 5 visualizes the capability-based task partitioning between the LLM validator (for $T_{reliable}$) and the human validator (for $T_{consistent}$) after the original SCICERO validation modules.

4.3. Human verification

A typical human verification workflow, where human participants judge the results of an automated approach, does not reduce the number of triples to be verified since all triples would be annotated both within the LLM validator module and the human validator module. Moreover, an over- or under-reliance on the LLM might affect the judgment capabilities of the human participants.

To address this issue, we adopt the "preference-testing" strategy from Faggioli et al. (2023), and apply it to the triple validation task such that a human judge is involved only whenever a disagreement among automated validation modules occurs.

Human validation upon correctness disagreement (workflow 5 in Table 1). The original SCICERO evaluation revealed that 34% of the triples removed by the transformer validator were in fact correct triples. In contrast, 35% of $T_{consistent}$ triples, added to the KG, were incorrect (Dessí et al., 2022a). To address this, as shown in Fig. 6, an additional LLM validator module can be integrated to re-evaluate triples before their removal or addition to the KG. The disagreement paradigm can be employed whenever the LLM validator produces an output different from the original SCICERO validation modules. Following the intuition that triples verified as correct by several distinct automated approaches are likely to be correct, human participants can focus only on annotating triples where no conclusive decision could be established.

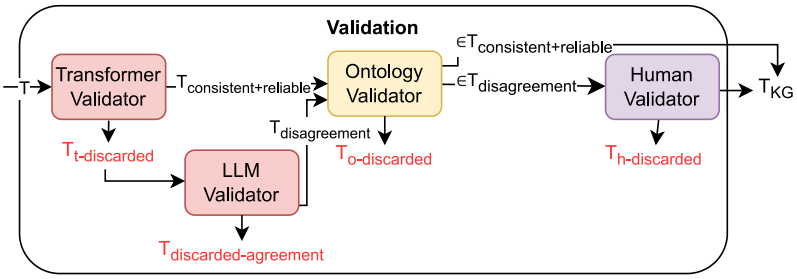

Fig. 7. Workflow 6: Human validation upon triple removal disagreement among the LLM validation and SCICERO transformer validator.

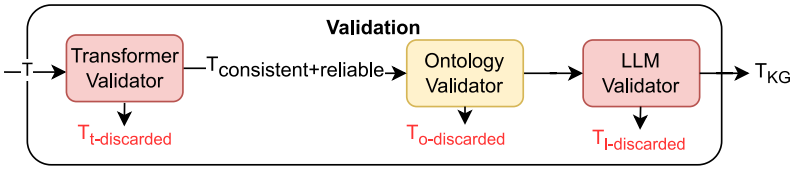

Fig. 8. Workflow 7: SCICERO integration with an LLM for triple addition approval.

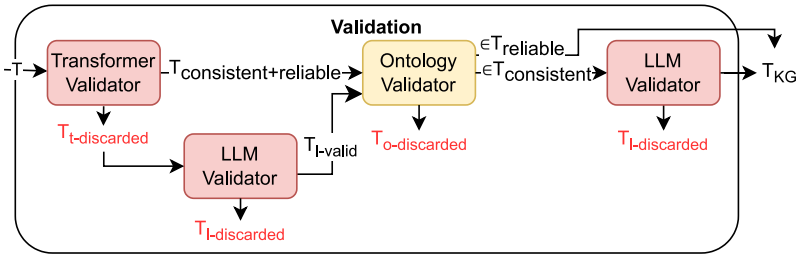

Fig. 9. Workflow 8: SCICERO integration with an LLM validator for triple addition approval of $T_{uncertain}$ triples.

Human validation upon triple removal disagreement (workflow 6 in Table 1). The previously proposed workflow can be adopted based on the main KG validation goals to reduce the manual efforts. For instance, workflow 6 (Fig. 7) follows the disagreement strategy only for the removal of triples and not for triple additions to the KG. Such an approach is especially useful whenever a higher KG coverage is desired. In comparison, for use cases with high KG precision requirements, such as those in the medical domain, the disagreement strategy might only be added for the addition of triples, to allow human participants to focus on removing misleading information.

4.4. Fully automated

We propose two integration approaches of the LLM module within the SCICERO framework.

SCICERO integration with the LLM validator (workflow 7 in Table 1). In this setup, the LLM validator is assumed to be better performing than the SCICERO automated validation and is therefore added as a last step of the workflow to ensure no noisy triples are included in the final KG. The transformer and ontology validators are utilized as filters reducing the LLM annotation costs, however, the final decision is taken by the LLM.

SCICERO with LLM validator approval for $T_{uncertain}$ (workflow 8 in Table 1). An alternative workflow, exploiting the task-partitioning paradigm, is shown in Fig. 9. The LLM validator module is integrated before the removal of triples or addition of triples with lower scientific support (i.e, triples belonging to $T_{uncertain}$). This workflow is the fully automated replication of workflow 5, however, instead of involving a HiL on disagreement, the annotation by the LLM is considered reliable and is used as the final decision.

Complete LLM validation (workflow 9 in Table 1). Building on related work in the field of LLMs (Chiang & Lee, 2023; Tsaneva, Vasic et al., 2024), we also propose a workflow in which the original SCICERO validation stage is entirely replaced by an LLM-based validator, as illustrated in Fig. 10. This workflow allows the investigation of the capabilities of LLMs as standalone annotators, compared to the performance achieved with the integration of the LLM within the SCICERO validation stage.

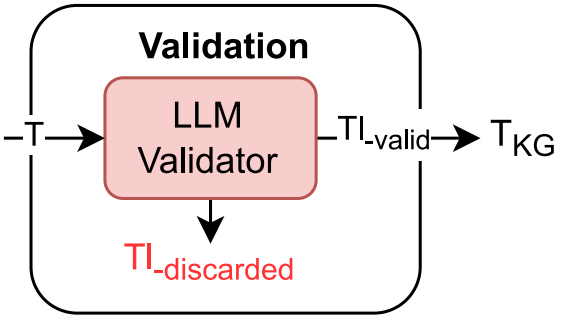

Fig. 10. Workflow 9: LLM validation in place of the original SCICERO validation.

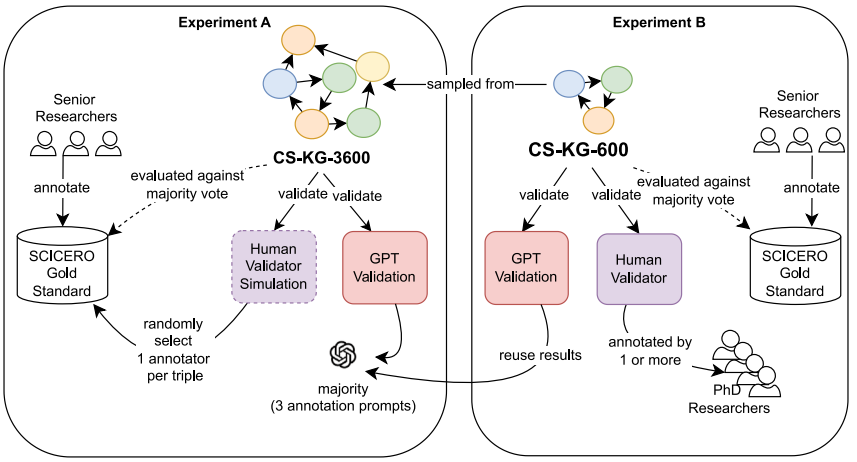

Fig. 11. Overview of the two experiment setups: Experiment A, conducted on the full CS-KG-3600 dataset by simulating the human validator module, and Experiment B, leveraging a subset CS-KG-600 and a concrete human validator implementation.

5. Experiment design and implementation

To evaluate each of the nine SCICERO validation workflows described in Section 4, we designed two experiments that differ in the dataset used and the implementation of the validator modules. It is important to emphasize that this study does not aim to evaluate the HiL or LLM validation modules in isolation. Instead, our objective is to assess their integration within an existing KG generation pipeline and to analyze the benefits of incorporating both LLMs and HiL in the validation stage. Therefore, we do not assess these modules separately using standard triple classification benchmarks.

An overview of the experimental investigation is illustrated in Fig. 11. We first conducted a large-scale simulation (Experiment A) using the adopted workflows to validate $3.6K$ triples. In this setting, the human validator module was simulated using the annotations of the original domain experts from the SCICERO evaluation (Dessí et al., 2022a), who manually classified each triple as either true or false. Our objective was to evaluate these collaborative workflows based on (1) precision, recall, and F1 scores, and (2) scalability, by analyzing the required number of HiL and LLM annotations. In Experiment B, we validated the findings of Experiment A by implementing an actual human validator module, comprising a pool of recruited domain experts. Given the additional manual effort required, this experiment was conducted on a representative subset of 600 triples from the original dataset.

We introduce the used dataset in Section 5.1 and describe the experimental setups of experiments A and B in Sections 5.2 and 5.3 respectively.

5.1. Experimental data

For the experimental investigations we make use of (1) CS-KG-3600: a gold standard created during the original SCICERO (Dessí et al., 2022a) evaluation,[5] and (2) CS-KG-600: a subset of this gold standard allowing for testing of additional validation implementations.

[5] The gold standard is available under https://github.com/danilo-dessi/SKG-pipeline/tree/main/eval

Table 2
Overview of the LLMs employed for the experiment, their version, and the date of the conducted experiment.

Model	Version	Experiment Date
GPT-4o	gpt-4o-2024-05-13	May 22nd, 2024
Claude Sonnet	claude-3-5-sonnet-20241022	Jan 31st, 2025
Llama 3.3 70B	Llama-3.3-70B-Instruct[a]	Feb 14, 2025

[a] We utilized the model available on Hugging Fac's platform accessible at https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

CS-KG-3600. For the SCICERO evaluation, a $3.6K$ triples set was sampled from the generated CS-KG, including equal amounts (600) of triples from each of the following categories:

- triples with very high support levels ($\in T_{reliable}$)
- triples with high support levels ($\in T_{reliable}$)
- triples with low support levels ($\in T_{consistent}$)
- triples labeled as incorrect by the transformer validator ($\in T_{t-discarded}$)
- triples removed by the ontology validator ($\in T_{o-discarded}$)
- randomly produced triples (T_{random}) generated by replacing the head or tail of CS-KG triples.

The triples were manually evaluated by 3 senior researchers in the Computer Science field and the annotations were aggregated using a majority voting strategy to determine the ground truth value for each triple. Further details are available in Dessí et al. (2022a).

CS-KG-600. We create a smaller representative dataset for exploration and findings validation purposes. CS-KG-600 is sampled from CS-KG-3600 and consists of 100 randomly selected triples from each of the six subsets contained within the gold standard.

5.2. Experiment A: Large scale simulation on CS-KG-3600

In this section, we describe the implementation of each newly added validation module for the first experiment (left-hand side of Fig. 11). This experiment was conducted on the full CS-KG-3600 gold standard.

LLM validator. The LLM module is implemented as follows:

- **Binary triple validation task.** Given a set of triples T, where each triple t is represented as $t = (subject, relation, object)$, the goal is to classify each triple as true or false, formalized as $V(t) : T \rightarrow \{0, 1\}$.
- **Employed LLMs.** We employed three alternative LLMs in our experiment: GPT-4o,[6] Claude Sonnet,[7] and Llama 3.3.[8] These models are among the highest-performing in the field and have each demonstrated superior performance across a wide range of tasks in previous studies. In particular, the GPT family has demonstrated excellent performance in class membership relation validation (Allen & Groth, 2024). Similarly, GPT-4o and Claude Sonnet have achieved expert-level ontology axiom validations, exceeding the results of open source models (Tsaneva, Herwanto et al., 2024).
  Additionally, we aimed to include an open model to accommodate scenarios where deploying a local solution is essential for validating private resources. While privacy concerns were not a factor in this study, since CS-KG exclusively extracts information from publicly available scholarly publications, we acknowledge that on-premise LLMs may be necessary in other applications. Since recent studies have shown that Llama 3 70B outperforms other open models in triple consistency validation (Regino & Dos Reis, 2025), we select its successor – Llama 3.3 – as on open source model for our experiments.
  Table 2 outlines the specific versions of the models employed in our study, along with the corresponding dates on which each experiment was performed.
- **LLM prompting.** Fig. 12 shows the initial instructions sent to the LLM model, introducing the annotation task and specifying the expected format of the response. Initial investigation revealed that requesting only the triple id and a binary judgment often resulted in incomplete or excessive judgments, which was addressed by requiring the complete triples to be included in the response.
  After the behavior of the LLM is defined, batches of 100 triples without any additional contextual information are sent for validation. In cases where the response did not match the required response format, the annotation was ignored and the same batch was sent again. To mimic a typical crowdsourcing experiment, we used the model's default parameters settings (e.g., temperature) and sent each batch of triples three consecutive times. The final judgment for each triple was determined using a majority vote aggregation of all responses.

[6] GPT-4o - https://openai.com/index/gpt-4o-system-card
[7] Claude Sonnet - https://www.anthropic.com/news/claude-3-5-sonnet
[8] Llama 3.3 - https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3

You are an expert in Computer Science and want to help with the identification of incorrect statements from the domain. The user will provide you with a set of RDF triples in the form (subject, predicate, object). For each triple from the set answer '0' if the statement they represent is incorrect and '1' if the modeled statement is correct. Think step by step when making the decision. Return the classifications of each triple in the order they were provided and do not add an explanation. Use the format
'0. [triple1]- [0|1], 1. [triple2]- [0|1],..., 99. [triple100]- [0|1]'.

Fig. 12. Initial instructions prompt sent to the LLM, introducing the annotation task, expected behavior and response format to be followed.

Human validator. Because of the large size of the dataset ($3.6K$ triples), to reduce the annotation efforts of the experiment, we simulate human-in-the-loop validations for the complete dataset by leveraging the original expert annotations used to create the gold standard. For each triple t, the gold standard dataset contains three binary expert annotations and the aggregated final ground truth value $A_t = \{a_1, a_2, a_3, a_{gt}\}$, $a_i \in \{0, 1\}$. For each triple t we select a single random expert judgment $a_r \in \{a_1, a_2, a_3\}$ to allow the reusability of the previously established gold standard while limiting biases introduced by the usage of the gold standard within the validation stage. Within the evaluation of the proposed SCICERO workflows, the selected random annotation a_r is compared against the ground truth value a_{gt}.

5.3. Experiment B: Real-life validation of CS-KG-600

The second experiment (right hand-side of Fig. 11) focuses on a realistic implementation of the human validator module. We perform the experiment on the subset CS-KG-600 to reduce additional manual annotation efforts.

LLM validator. For the LLM validator implementation we reuse the produced LLM annotations previously described in Section 5.2. Since CS-KG-600 is sampled from CS-KG-3600, no additional LLM annotation had to be carried out.

Human validator.

- **Binary triple validation task.** Given a set of triples T, where each triple t is represented as $t = (subject, relation, object)$, and triple context $C_t = (subject\_type, object\_type, file\_ids)$, containing the types of the subject and object nodes as well the identifiers of the files where the triples were extracted from. The goal is to classify each triple as true or false, formalized as $V(t, Ct) : T \rightarrow \{0, 1\}$.
- **Sample size.** To minimize the annotation efforts, only triples which are sent to the human validator module in one of the workflows 2–9 are annotated. In total, 333 triples were reviewed.
- **Annotators background.** The annotators were four advanced PhD researchers with Computer Science (or equivalent) background, who did not have any involvement in the creation of CS-KG, SCICERO or the initial gold standard creation. They were asked to judge whether a triple is correct or incorrect according to their expertise and support in scientific literature.
- **Annotation environment.** The annotations were performed using Google Sheets, where each triple was presented in natural language, displaying the subject, object, and relation, along with the types of the subject and object. During the annotation process, participants had access to the files associated with the articles from which each triple was extracted and could use digital libraries such as Scopus[9] to browse through additional scientific content in order to provide an informed decision.
  Example triples provided to the annotators with contextual information are shown in Table 3. The first columns represent the triple elements. Let us consider the validation of the triple <*natural language processing, uses, word segmentation*> as an example. As contextual information, the annotators are provided with the knowledge that both *natural language processing* and *word segmentation* are classified as Task. Additionally, they are given the identifiers of three files from which the triple was extracted. To access the original papers, they were instructed to use OpenAlex.[10] For instance, the file identified as **2110300018** can be accessed at https://openalex.org/works/w2110300018.
- **Annotation strategy.** Each triple was first annotated by a single expert. In cases where the expert had doubts about the correctness of a triple, a second expert was asked for a judgment. If the annotations were inconclusive, a discussion among the two experts took place and if no agreement could be reached, a third expert was involved and the final decision was reached through majority vote. Each annotator worked at their own pace, and, following the necessary discussions, the final annotations were completed within a week.

[9] Scopus - https://www.scopus.com/
[10] OpenAlex - https://openalex.org/

Table 3
Example triples and the provided context in the format presented to the annotators.

Subject	Relation	Object	Subject type	Object type	Files
natural language processing	uses	word segmentation	Task	Task	2110300018, 158142204, 2146654604
information integration	acquires	patient address	Task	Other Entity	2849896015
semantic profile representation	acquires	document classification	Method	Task	2039759410

Table 4
Experiment A results on the CS-KG-3600 dataset in terms of precision (P), recall (R), F1 scores and additional resource efforts provided as N_{LLM} and N_{Human}. The best score for each workflow across LLMs is shown in **bold** and the best scores per workflow type are underlined.

Workflow ID	Workflow Fig.	Model	Performance P	Performance R	Performance F1	Annotations N_{LLM}	Annotations N_{Human}
SCICERO							
	Fig. 1	–	75%	79%	77%	–	–
Human Judgment							
1	Fig. 2	–	88%	87%	87%	–	3.6K
2	Fig. 3	–	93%	71%	81%	–	1.8K
3	Fig. 4	–	83%	77%	80%	–	600
AI Assistance							
4	Fig. 5	gpt-4o	**90%**	66%	76%	1.2K	600
		claude sonnet	87%	69%	77%	1.2K	600
		llama 3.3 70B	85%	**73%**	**79%**	1.2K	600
Human Verification							
5	Fig. 6	gpt-4o	**80%**	82%	81%	1.1K	423
		claude sonnet	78%	**86%**	**82%**	1.1K	444
		llama 3.3 70B	78%	**86%**	**82%**	1.1K	442
6	Fig. 7	gpt-4o	75%	83%	79%	505	158
		claude sonnet	75%	**87%**	**81%**	505	324
		llama 3.3 70B	75%	86%	80%	505	331
Full Automation							
7	Fig. 8	gpt-4o	**87%**	62%	72%	1.8K	–
		claude sonnet	81%	69%	75%	1.8K	–
		llama 3.3 70B	80%	**73%**	**76%**	1.8K	–
8	Fig. 9	gpt-4o	**78%**	77%	77%	1.1K	–
		claude sonnet	74%	**87%**	**80%**	1.1K	–
		llama 3.3 70B	72%	86%	79%	1.1K	-
9	Fig. 10	gpt-4o	**70%**	70%	70%	3.6K	–
		claude sonnet	61%	85%	**71%**	3.6K	–
		llama 3.3 70B	58%	**89%**	70%	3.6K	–

6. Results

In this section, we present the results achieved with the nine validation workflows in each of the conducted experiments.

6.1. Experiment A - Results of the large scale simulation on CS-KG-3600

For the first experiment, we utilized the SCICERO complete gold standard to simulate the proposed extended SCICERO workflows and implemented the human validator module such that a random expert annotation from the ground truth is used within the workflows. Table 4 shows an overview of the workflow performance in terms of precision, recall and F1 scores as well as additional efforts added to SCICERO, i.e., the amount of triples to undergo an LLM (N_{LLM}) or human (N_{Human}) validation. The scores are color-coded for an easy overview of the improvements (in green) and losses (in red), introduced by each workflow with respect to the original SCICERO performance on the datasets. We discuss our results along the four categories of workflows from *Human Judgment* to *Full Automation*.

Notably, the *human judgment* workflows offer precision improvements of 8%–18% with possible recall losses (up to 8%). While the F1 improvements are prominent for this workflow type, the number of triples to be manually evaluated, especially for workflows 1 and 2, are considerably high ($\geq 1.8K$) and as such, these workflows are only suitable for small-size resources. For workflow 3, the

Table 5
Experiment B results on the CS-KG-600 dataset in terms of precision (P), recall (R), F1 scores and additional resource efforts provided as N_{LLM} and N_{Human}. The best score for each workflow across LLMs is shown in **bold** and the best scores per workflow type are underlined. Scores marked with a star (∗) are estimated based on a subset annotation.

Workflow ID	Workflow Fig.	Model	Performance P	Performance R	Performance F1	Annotations N_{LLM}	Annotations N_{Human}
SCICERO							
	Fig. 1	–	73%	77%	75%	–	–
Human Judgment							
1	Fig. 2		77%*	55%*	64%*	–	600
2	Fig. 3		81%	42%	56%	–	300
3	Fig. 4		77%	67%	72%	–	100
AI Assistance							
4	Fig. 5	gpt-4o	**84%**	57%	68%	200	100
		claude sonnet	81%	57%	67%	200	100
		llama 3.3 70B	80%	**63%**	**71%**	200	100
Human Verification							
5	Fig. 6	gpt-4o	**77%**	77%	77%	178	72
		claude sonnet	75%	**83%**	**79%**	178	75
		llama 3.3 70B	74%	82%	78%	178	61
6	Fig. 7	gpt-4o	73%	80%	76%	78	26
		claude sonnet	73%	**84%**	78%	78	50
		llama 3.3 70B	73%	83%	78%	78	46
Full Automation							
7	Fig. 8	gpt-4o	**85%**	59%	70%	300	–
		claude sonnet	80%	63%	71%	300	–
		llama 3.3 70B	78%	**71%**	**74%**	300	–
8	Fig. 9	gpt-4o	**78%**	76%	77%	178	–
		claude sonnet	74%	**85%**	**79%**	178	–
		llama 3.3 70B	72%	84%	77%	178	–
9	Fig. 10	gpt-4o	**68%**	69%	**69%**	600	–
		claude sonnet	61%	81%	**69%**	600	–
		llama 3.3 70B	56%	**87%**	68%	600	–

amount of manual annotations was significantly reduced by leveraging the capabilities of the SCICERO validators to filter reliable triples. While the precision is lowest compared to other workflows of this type, the F1 score is still increased to 80%, similarly to workflow 2 (81%), in which three times more triples are annotated.

In the *AI assistance* category, workflow 4, which extended the workflow 3 manual validation with an additional LLM validator module, further improves the precision to 90% (+15% from the SCICERO baseline) when the GPT-4o model is used. Nevertheless, the loss in recall (−13%) results in a small F1 score decrease (−1%). As such, this workflow is suitable for evaluation campaigns where a high precision of the KG is required, while the KG coverage is not a main priority.

The workflows from the *human verification* interaction-level type lead to score improvements in all performance scores while keeping the human annotations to a minimum (< 13% of the total triples). Workflow 5 improves the precision by up to 5% by utilizing GPT-4o while workflow 6 reaches up to 8% recall increase when using Claude Sonnet.

The *fully automated* workflows showcase the capabilities of the LLM validator module. Workflow 7 and 8 manage to increase the precision by 3%–12% with no additional manual efforts when using GPT-4o. Workflow 9, which relies solely on LLM-based validation, is the only workflow that leads to an overall decrease in precision and F1 scores. This outcome highlights the crucial role of integrating LLMs with other automated approaches to enhance performance.

The experiment highlighted the strengths and limitations of different automation levels that integrate LLMs with human-in-the-loop approaches. While *human-judgment* workflows yield the most significant improvements over the original SCICERO pipeline, they are labor-intensive and poorly suited for evaluating large-scale knowledge graphs. At the other end of the spectrum, the workflows based on *full automation* exhibited the highest scalability, albeit at the cost of some performance. In contrast, *AI Assistance* workflows can partially address scalability challenges and enhance precision but suffer from low recall. The *human-verification* approach seems to offer the best trade-off between performance and manual effort, as it effectively improves overall results while minimizing human intervention.

6.2. Experiment B - Results of the real-life validation of CS-KG-600

For the second experiment, we utilized CS-KG-600, a subset of the SCICERO gold standard, and conducted an additional human-in-the-loop annotation campaign. Table 5 provides a overview of the performance achieved by the various workflows.

Similarly to Experiment A, the *human judgment* workflows increase the precision results compared to the SCICERO baseline (+ 4%–8%). However, in contrast to the results from Experiment A, the added benefit is not as high and the losses in recall and F1

score are significant (up to −35%). Performance decreases were expected since in Experiment A the ground truth was used within the simulated workflows. These results can further be explained by the domain expertise level of the involved human participants (senior vs junior experts). However, further investigations are needed to explore other factors, which may have influenced the scores.

It should be noted that because of the high manual efforts in the validation following workflow 1, we do not test this entire workflow in Experiment B. The values in Table 5 are estimations based on the annotations performed for the remaining workflows (56% of the CS-KG-600 dataset).

Surprisingly, the *fully automated* and *AI-assisted* workflows result in a higher increase in precision (+ 5%–12% when using GPT-4o), compared to the performance obtained using human judgment. As with the previous experiment, there is a trade-off with the recall scores which decrease by up to 20%. An exception is the implementation of workflow 8 with the Claude Sonnet model, which leads to overall improved performance scores. However, these effects are not consistently observed across models, and further investigation is needed to determine whether similar performance can be reliably replicated.

Human verification workflows, as in Experiment A, lead to balanced performance improvements across all tested LLMs. The increase is rather small (+4% precision in workflow 4; +3% recall in workflow 5). However, it is noteworthy that these gains come without any trade-offs in other performance metrics.

The optimal workflow should be selected based on the main evaluation objective and the available resources. As demonstrated in our experiments, precision scores can be enhanced both with and without additional manual intervention. While fully manual or fully automated workflows inherently involve a trade-off between precision and recall, human-in-the-loop verification strategies effectively improve overall performance while minimizing manual effort.

7. Conclusion

The automated generation of knowledge graphs enables extensive content coverage of the represented domains. However, such automatically curated resources often contain quality issues. To ensure the quality of the generated KGs and the success of the applications relying on them, validation is a crucial step in the generation process. This paper explores innovative validation approaches for knowledge graphs that combine human-in-the-loop techniques with LLMs. Specifically, we investigate potential workflows that integrate LLMs and human contributors at varying levels of automation (RQ1) and evaluate their strengths and limitations (RQ2) in terms of both performance metrics (precision, recall, and F1 score) and scalability.

We validate the Computer Science Knowledge Graph (CS-KG) as a use case. CS-KG is a valuable resource that integrates scientific claims from millions of publications, facilitating the analysis of research trends and supporting various scientometric tasks (Dessí et al., 2022b). Thus, it is fundamental to ensure the quality of the knowledge graph by removing misleading or incorrectly extracted statements. To this end, we extend SCICERO – the pipeline used to produce CS-KG – by integrating LLMs and HiL within its validation stage.

This section presents a summary of the key contributions and findings of our research. Additionally, we discuss the limitations, propose directions for future research, and outline open issues and remaining challenges.

Contributions. We extend SCICERO by integrating additional LLMs and human-based validator modules and make the following contributions to the field:

- **Human-LLM collaboration investigation.** We present an overview of possible SCICERO extensions, incorporating LLMs and/or human-in-the-loop validation techniques. We explore a spectrum of collaboration levels, raging from fully manual human judgments and AI-assisted annotations to human verification and full automation. This results in nine different workflows combining HiL and LLMs.
- **LLM-based KG validation.** We propose a concrete implementation of an LLM-sourced KG triple annotation and asses the achieved validation performance of three LLMs.
- **Experimental evaluation.** We conduct two experimental investigations using the SCICERO gold standard, consisting of $3.6K$ triples, to empirically evaluate the strength and limitations of each of the nine proposed collaborative validation workflows. As such, we empirically shed light on the trade-offs of various human-LLM combination possibilities. To the best of our knowledge, we pioneer the empirical exploration of the trade-offs of such hybrid workflows on large-scale datasets for the task of KG validation.
- **Annotation collection.** We publish the collected annotations produced in the LLM and HiL validation modules online[11] to allow the reproducibility of our work as well as the exploration of further workflows and in-depth experiments by fellow researchers.

Main findings. Our experimental investigation yields the following key insights, relevant to the community:

- **Weak performance of standalone LLM validation.** LLMs, when used independently, fail to deliver highly accurate KG triple validations (up to 70% precision, workflow 9). However, when combined with other automated validation methods, as in workflows 7 & 8, precision improves significantly, reaching up to 87%.

[11] Knowledge Graph Triple Validation by LLMs and Human-in-the-Loop [Data set]. Zenodo. https://doi.org/10.5281/zenodo.13730203

- **Human-level validation by integrated LLMs.** The integration of LLMs into the SCICERO pipeline produces results comparable to, and in some cases better than, SCICERO workflows, including a human-in-the-loop. Concretely, in Experiment B, workflow 7 achieves 85% precision and outperforms the human judgment workflows by 4%–8%.
- **Superior hybrid human-LLM collaboration.** The integration of LLMs and human annotators successfully balances precision and recall, overcoming trade-off limitations observed in workflows that rely solely on either human expertise or LLM-sourced annotations. These improvements are shown by workflows 5 & 6 in both experimental setups across tested LLMs.
- **LLM disagreement strategy for HiL involvement.** A promising method for reducing human intervention is the disagreement strategy employed by workflows 5 & 6: when two automated validators produce inconsistent annotations, human contributors can resolve the conflict. If both automated methods agree, their validation can be considered reliable and manual checks can be avoided.

Limitations and outlook. Despite the valuable insights gained in this study, several limitations and challenges remain:

- **Single use case.** This study focuses on a single KG generation pipeline –SCICERO– to enable a detailed and in-depth analysis of a concrete use case. Since several components of SCICERO are specifically designed for the Computer Science domain, the implementation developed for this study cannot be directly applied to other domains. However, the theoretical framework explored in this study, along with the nine workflows, was designed to be domain-independent and can be readily implemented in various fields with minimal or no modifications. Similarly, the insights gained from our study can inform the development of novel systems that integrate HiL approaches and LLMs. Furthermore, even the specific framework modeled on CS-KG can be adapted to other domains by adjusting the extraction components and incorporating a relevant domain ontology. Nevertheless, such adaptations require additional domain expertise and evaluation efforts, which fall beyond the scope of this study. To further validate and generalize our findings, we plan to conduct a series of follow-up experiments to assess the adaptability of the proposed workflows across different domains and KG generation solutions.
- **Further SCICERO integration.** Future work will involve scaling up the evaluation by applying the validation pipelines to a larger subset of CS-KG to further validate the findings of this study. Moreover, we intend to utilize the extended SCICERO pipelines to generate different versions of CS-KG, allowing for a cross validation and analysis of tasks enabled by the KG such as forecasting of research dynamics.
- **LLM (prompt) variability.** While our implementation, which utilized three different LLMs, produces human-level annotations, further studies are necessary to assess the impact of prompt modifications and model selection on overall performance. Future work could explore workflows that integrate retrieval-augmented generation (RAG) and evaluate how relevant contextual information impacts performance.
- **Scalability.** Although the hybrid human-LLM SCICERO extensions reduce manual efforts significantly, scalability remains a challenge for large resources containing millions of triples. Further investigations will explore ways to extend SCICERO with other strategies such as triple annotation priority to prevent bottlenecks while maintaining high validation quality. We also plan to investigate an experimental setup that integrates multiple SCICERO workflows. This approach would enable dynamic selection of the most suitable workflow based on real-time assessments of both human and LLM resource availability.
- **Evaluating LLM annotations.** In this paper, LLM annotations have solely been compared against human-generated annotations. However, our experiments indicate that LLMs can outperform junior experts. Further research is needed to re-evaluate current ground truth creation methods and explore new measurements to detect cases when LLMs exceed human expertise.

Validating semantic resources using LLMs is a complex challenge. In this work, we contribute to the field by proposing and evaluating various workflows that incorporate HiL methods, LLMs, or a combination of both, applied to a large-scale resource. Our findings underscore the strengths and limitations of each approach, demonstrating the potential of LLMs as a valuable complement for generating high-quality knowledge graphs at scale. Additionally, our insights into hybrid workflows integrating HiL and LLMs are relevant to researchers working on other knowledge-intensive tasks beyond KG validation.

CRediT authorship contribution statement

Stefani Tsaneva: Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Danilo Dessì:** Writing – review & editing, Data curation, Conceptualization. **Francesco Osborne:** Writing – review & editing, Data curation, Conceptualization. **Marta Sabou:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT4 in order to suggest improvements to the readability and language of the manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank all data annotators for their involvement and contributions.

This research was funded in whole or in part by the Austrian Science Fund (FWF) BILAI (10.55776/COE12) and HOnEst (V 745) projects. For open access purposes, the author has applied a CC BY public copyright license to any author accepted manuscript version arising from this submission. Additionally, the work was supported by the PERKS (101120323) project, co-funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Data availability

The SCICERO gold standard and collected annotations from the experiments are available at https://github.com/danilo-dessi/SKG-pipeline/blob/main/eval/ and https://doi.org/10.5281/zenodo.13730203.

References

Allen, B. P., & Groth, P. T. (2024). Evaluating class membership relations in knowledge graphs using large language models. In *The semantic web: ESWC satellite events*.

Alonso, R., Dessí, D., Meloni, A., & Reforgiato Recupero, D. (2025). A novel approach for job matching and skill recommendation using transformers and the o*net database. *Big Data Research*, *39*, Article 100509. http://dx.doi.org/10.1016/j.bdr.2025.100509.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, *26*.

Borrego, A., Ayala, D., Hernández, I., Rivero, C. R., & Ruiz, D. (2021). CAFE: Knowledge graph completion using neighborhood-aware features. *Engineering Applications of Artificial Intelligence*, *103*, Article 104302.

Borrego, A., Dessì, D., Hernández, I., Osborne, F., Recupero, D. R., Ruiz, D., et al. (2022). Completing scientific facts in knowledge graphs of research concepts. *IEEE Access*, *10*, 125867–125880.

Carta, S., Giuliani, A., Piano, L., Podda, A. S., Pompianu, L., & Tiddia, S. G. (2023). Iterative zero-shot LLM prompting for knowledge graph construction. arXiv preprint arXiv:2307.01128.

Chessa, A., Fenu, G., Motta, E., Osborne, F., Recupero, D. R., Salatino, A., et al. (2023). Data-driven methodology for knowledge graph generation within the tourism domain. *IEEE Access*, *11*, 67567–67599.

Chiang, C.-H., & Lee, H.-y. (2023). Can large language models be an alternative to human evaluations? In *Proceedings of the 61st annual meeting of the association for computational linguistics*. http://dx.doi.org/10.18653/v1/2023.acl-long.870.

Demartini, G., Difallah, D. E., & Cudré-Mauroux, P. (2013). Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *VLDB Journal*, *22*(5), 665–687. http://dx.doi.org/10.1007/s00778-013-0324-z.

Dessí, D., Osborne, F., Recupero, D. R., Buscaldi, D., & Motta, E. (2022a). SCICERO: A deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. *Knowledge-Based Systems*, *258*, http://dx.doi.org/10.1016/j.knosys.2022.109945.

Dessí, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., & Motta, E. (2022b). CS-KG: A large-scale knowledge graph of research entities and claims in computer science. In *The semantic web – ISWC 2022* (pp. 678–696). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-19433-7{_}39.

Dessì, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., Motta, E., & Sack, H. (2020). AI-KG: an automatically generated knowledge graph of artificial intelligence. In *The semantic web–ISWC 2020: 19th international semantic web conference, proceedings, part II 19* (pp. 127–143). Springer, http://dx.doi.org/10.1007/978-3-030-62466-8{_}9.

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32*.

Faggioli, G., Dietz, L., Clarke, C. L. A., Demartini, G., Hagen, M., Hauff, C., et al. (2023). Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR international conference on theory of information retrieval* (pp. 39–50). Association for Computing Machinery, http://dx.doi.org/10.1145/3578337.3605136.

Fathallah, N., Das, A., De Giorgis, S., Poltronieri, A., Haase, P., & Kovriguina, L. (2024). NeOn-GPT: A large language model-powered pipeline for ontology learning. In *The semantic web: ESWC satellite events*.

Gao, J., Li, X., Xu, Y. E., Sisman, B., Dong, X. L., & Yang, J. (2019). Efficient knowledge graph accuracy evaluation. *Proceedings of the VLDB Endowment*, *12*(11), 1679–1691. http://dx.doi.org/10.14778/3342263.3342642.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., et al. (2020). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, *34*(8), 3549–3568. http://dx.doi.org/10.1109/TAEAC50856.2021.9390863.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., et al. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4), http://dx.doi.org/10.1145/3447772.

Hoppe, F., Dessì, D., & Sack, H. (2021). Understanding class representations: An intrinsic evaluation of zero-shot text classification. In M. Alam, D. Buscaldi, M. Cochez, D. R. Recupero, & H. Sack (Eds.), *CEUR workshop proceedings*: *Vol. 3034*, *Proceedings of the workshop on deep learning for knowledge graphs (DL4KG 2021) co-located with the 20th international semantic web conference*.

Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., et al. (2019). Open Research Knowledge Graph: Next generation infrastructure for semantic scholarly knowledge. In *K-CAP '19*, *Proceedings of the 10th international conference on knowledge capture* (pp. 243–246). Association for Computing Machinery, http://dx.doi.org/10.1145/3360901.3364435.

Jaradeh, M. Y., Singh, K., Stocker, M., & Auer, S. (2021). Triple classification for scholarly knowledge graph completion. In *Proceedings of the 11th knowledge capture conference* (pp. 225–232).

Khorashadizadeh, H., Mihindukulasooriya, N., Tiwari, S., Groppe, J., & Groppe, S. (2023). Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text. In *TEXT2KG/biKE@eSWC*. arXiv:2305.08804.

Kovacevic, F., El-Ebshihy, A., Piroi, F., & Rauber, A. (2024). Extending content-based scientific knowledge graphs with research results. In *KG-neSy workshop, co-located with the the 1st Austrian symposium on AI, robotics, and vision (AIROV24)*.

Li, X., & Daoutis, M. (2021). Unsupervised key-phrase extraction and clustering for classification scheme in scientific publications. In A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, & L. A. Celi (Eds.), *CEUR workshop proceedings*: *Vol. 2831*, *Proceedings of the workshop on scientific document understanding co-located with 35th AAAI conference on artificial inteligence*.

Li, L., Wang, P., Yan, J., Wang, Y., Li, S., Jiang, J., et al. (2020). Real-world data medical knowledge graph: construction and applications. *Artificial Intelligence in Medicine*, *103*, Article 101817. http://dx.doi.org/10.1016/j.artmed.2020.101817.

Li, C., Zhao, P., Sheng, V. S., Xian, X., Wu, J., & Cui, Z. (2017). Refining automatically extracted knowledge bases using crowdsourcing. *Computational Intelligence and Neuroscience*, *2017*, 1–17. http://dx.doi.org/10.1155/2017/4092135.

Lossio-Ventura, J. A., Hogan, W., Modave, F., Guo, Y., He, Z., Yang, X., et al. (2018). OC-2-KB: Integrating crowdsourcing into an obesity and cancer knowledge base curation system. *BMC Medical Informatics and Decision Making*, *18*, http://dx.doi.org/10.1186/s12911-018-0635-5.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd annual meeting of the association for computational linguistics, ACL* (pp. 55–60). The Association for Computer Linguistics, http://dx.doi.org/10.3115/V1/P14-5010.

Meloni, A., Angioni, S., Salatino, A., Osborne, F., Recupero, D. R., & Motta, E. (2023). Integrating conversational agents and knowledge graphs within the scholarly domain. *Ieee Access*, *11*, 22468–22489.

Nickel, M., Tresp, V., & Kriegel, H.-P. (2012). Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on world wide web* (pp. 271–280).

Nuzzolese, A. G., Presutti, V., Gangemi, A., Peroni, S., & Ciancarini, P. (2017). Aemoo: Linked data exploration based on knowledge patterns. *Semantic Web*, *8*(1), 87–112. http://dx.doi.org/10.3233/SW-160222.

Ojha, P., & Talukdar, P. (2017). Kgeval: Accuracy estimation of automatically constructed knowledge graphs. In *Proc. conf. empirical methods in natural language processing* (pp. 1741–1750). http://dx.doi.org/10.18653/v1/d17-1183.

Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web Journal*, *8*(3), 489–508. http://dx.doi.org/10.3233/SW-160218.

Peng, C., Xia, F., Naseriparsa, M., & Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, *56*, 1–32. http://dx.doi.org/10.1007/s10462-023-10465-9.

Pomp, A., Lipp, J., & Meisen, T. (2019). You are missing a concept! Enhancing ontology-based data access with evolving ontologies. In *IEEE 13th int. conf. semantic computing* (pp. 98–105). http://dx.doi.org/10.1109/ICOSC.2019.8665620.

Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). OOPS!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *10*(2), 7–34. http://dx.doi.org/10.4018/ijswis.2014040102.

Qi, Y., Zheng, W., Hong, L., & Zou, L. (2022). Evaluating knowledge graph accuracy powered by optimized human-machine collaboration. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1368–1378). Association for Computing Machinery, http://dx.doi.org/10.1145/3534678.3539233.

Regino, A. G., & Dos Reis, J. C. (2025). Can LLMs be knowledge graph curators for validating triple insertions? In *Proceedings of the workshop on generative AI and knowledge graphs (GenAIK)* (pp. 87–99).

Rumin1, G., & Mekterović, I. (2019). LOD construction through SupervisedWeb relation extractionand crowd validation. *Journal of Web Engineering*, *18*(1), 229–256. http://dx.doi.org/10.13052/jwe1540-9589.18137.

Sakor, A., Jozashoori, S., Niazmand, E., Rivas, A., Bougiatiotis, K., Aisopos, F., et al. (2023). Knowledge4COVID-19: A semantic-based approach for constructing a COVID-19 related knowledge graph from various sources and analyzing treatments' toxicities. *Journal of Web Semantics*, *75*, Article 100760. http://dx.doi.org/10.1016/j.websem.2022.100760.

Salatino, A., Mannocci, A., Osborne, F., Rehm, G., Schimmler, S., et al. (2024). 4Th international workshop on scientific knowledge: Representation, discovery, and assessment. In *CEUR WORKSHOP pROCEEDINGS, vol. 3780*. CEUR-WS.

Salatino, A. A., Osborne, F., & Motta, E. (2022). CSO Classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics. *International Journal on Digital Libraries*, *23*(1), 91–110. http://dx.doi.org/10.1007/S00799-021-00305-Y.

Sallam, M., Al-Salahat, K., Eid, H., Egger, J., & Puladi, B. (2024). Human versus artificial intelligence: Chatgpt-4 outperforming bing, bard, ChatGPT-3.5, and humans in clinical chemistry multiple-choice questions. *MedRxiv*, http://dx.doi.org/10.1101/2024.01.08.24300995.

Su, Y., & Zhang, Y. (2020). Automatic construction of subject knowledge graph based on educational big data. In *Proceedings of the 2020 3rd international conference on big data and education* (pp. 30–36). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3396452.3396458.

Trajanoska, M., Stojanov, R., & Trajanov, D. (2023). Enhancing knowledge graph construction using large language models. arXiv:2305.04676.

Tsaneva, S., Dessì, D., Osborne, F., & Sabou, M. (2024). Enhancing scientific knowledge graph generation pipelines with LLMs and human-in-the-loop. In *4th international workshop on scientific knowledge: representation, discovery, and assessment, co-located with the 23rd international semantic web conference, ISWC 2024*.

Tsaneva, S., Herwanto, G., & Sabou, M. (2024). Benchmarking ontology validation capabilities of LLMs. In *The 23rd international semantic web conference 2024, special session on harmonising generative AI and semantic web technologies*.

Tsaneva, S., Vasic, S., & Sabou, M. (2024). LLM-driven ontology evaluation: Verifying ontology restrictions with ChatGPT. In *The semantic web: ESWC satellite events*.

van Stijn, J. J., Neerincx, M. A., ten Teije, A., & Vethman, S. (2021). Team design patterns for moral decisions in hybrid intelligent systems: A case study of bias mitigation. In *CEUR workshop proceedings, vol. 2846*. CEUR-WS.

Wang, Y., Cheng, Y., Qi, Q., & Tao, F. (2024). IDS-KG: An industrial dataspace-based knowledge graph construction approach for smart maintenance. *Journal of Industrial Information Integration*, *38*, Article 100566. http://dx.doi.org/10.1016/j.jii.2024.100566.

Xu, Z., & Dang, Y. (2023). Data-driven causal knowledge graph construction for root cause analysis in quality problem solving. *International Journal of Production Research*, *61*(10), 3227–3245. http://dx.doi.org/10.1080/00207543.2022.2078748.

Xue, B., & Zou, L. (2023). Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4969–4988. http://dx.doi.org/10.1109/TKDE.2022.3150080.

Yani, M., & Krisnadhi, A. A. (2021). Challenges, techniques, and trends of simple knowledge graph question answering: a survey. *Information*, *12*(7), 271. http://dx.doi.org/10.3390/info12070271.

Yao, L., Mao, C., & Luo, Y. (2019). KG-BERT: BERT for knowledge graph completion. arXiv preprint arXiv:1909.03193.

Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S., et al. (2013). Quality assessment methodologies for linked open data. *Semantic Web Journal*, *1*(1), 1–5. http://dx.doi.org/10.3233/SW-150175.

Zhang, B., Reklos, I., Jain, N., Peñuela, A. M., & Simperl, E. (2023). Using large language models for knowledge engineering (LLMKE): A case study on wikidata. arXiv preprint arXiv:2309.08491.

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., et al. (2024). LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web*, *27*, http://dx.doi.org/10.1007/s11280-024-01297-w.