Enhancing Scientific Knowledge Graph Generation Pipelines with LLMs and Human-in-the-Loop

Stefani Tsaneva[1,*], Danilo Dessì[2,†], Francesco Osborne[3,4,†] and Marta Sabou[1]

[1] *Institute of Data, Process and Knowledge Management, Vienna University of Economics and Business, Austria*
[2] *Computer Science Department, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates*
[3] *Knowledge Media Institute, The Open University, United Kingdom*
[4] *Department of Business and Law, University of Milano Bicocca, Italy*

Abstract

Scientific Knowledge Graphs have recently become a powerful tool for exploring the research landscape and assisting scientific inquiry. It is crucial to generate and validate these resources to ensure they offer a comprehensive and accurate representation of specific research fields. However, manual approaches are not scalable, while automated methods often result in lower-quality resources. In this paper, we investigate novel validation techniques to improve the accuracy of automated KG generation methodologies, leveraging both a human-in-the-loop (HiL) and a large language model (LLM)-in-the-loop. Using the automated generation pipeline of the Computer Science Knowledge Graph as a case study, we demonstrate that precision can be increased by 12% (from 75% to 87%) using only LLMs. Moreover, a hybrid approach incorporating both LLMs and HiL significantly enhances both precision and recall, resulting in a 4% increase in the F1 score (from 77% to 81%).

Keywords

Knowledge Graph Evaluation, Scientific Knowledge Graph, Large Language Models, Hybrid Human-AI Workflows

1. Introduction

As the number of open research articles continues to grow, the research community increasingly needs efficient solutions for knowledge-based content exploration of scientific works. Knowledge graphs (KGs) have emerged as a crucial technology in this domain due to their ability to structure information semantically and support intelligent systems [1]. Consequently, scientific KGs, which facilitate the categorization, search, and reasoning over scientific knowledge, have attracted significant interest (e.g., [2, 3, 4, 5, 6, 7, 8]). Some of these resources, such as the Open Research Knowledge Graph (ORKG) [2], require manual annotations for their creation. This approach produces high-quality data, but it limits the coverage and scalability of the curated resources. Other methods aim to generate much larger resources by integrating scientific content from millions of articles through automated processes. An example is the Computer Science Knowledge Graph (CS-KG) [5], a KG of 10 million entities extracted from 6.7 million publications built using an automatic pipeline called SCICERO [4]. However, while fully automated KG generation approaches provide extensive coverage of the represented area, they often fall short in terms of the quality of the resulting KGs. Due to the complexity of transforming natural language in structured form for example using the Resource Description Framework (RDF) misleading or incorrect triples might be extracted. Therefore, it is crucial to incorporate validation steps as part of the scientific KG generation and curation processes. To this aim, SCICERO includes a module to assess triples according to domain understanding through an ontology, and a module that assesses triples using the sci-bert [9] model. However, SCICERO does not make use of common

Sci-K'24: 4th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment, November 2024, Baltimore, MD, USA, co-located with The 23rd International Semantic Web Conference, ISWC 2024

[*] Corresponding author.

[†] These authors contributed equally.

stefani.tsaneva@wu.ac.at (S. Tsaneva); ddessi@sharjah.ac.ae (D. Dessì); francesco.osborne@open.ac.uk (F. Osborne); marta.sabou@wu.ac.at (M. Sabou)

0000-0002-0895-6379 (S. Tsaneva); 0000-0003-3843-3285 (D. Dessì); 0000-0001-6557-3131 (F. Osborne); 0000-0001-9301-8418 (M. Sabou)

methods based on human-in-the-loop (HiL) approaches, where domain experts identify incorrect triples to improve the quality of the resulting KGs. Traditional HiL methods, however, are not scalable [10], thus making challenging their use for large KGs. A potential solution is the adoption of modern large language models (LLMs), which have demonstrated human-like performance in many natural language processing tasks [11].

In this paper, we investigate novel validation approaches to improve the accuracy of automated KG generation pipelines, leveraging both a HiL and an *LLM-in-the-loop* (LLM-iL). Using SCICERO- the automated generation pipeline of the Computer Science Knowledge Graph as a case study [4], we design and test validation modules that can be integrated into the existing pipeline to improve the quality of the resulting triples, leading to an enhanced CS-KG. For this purpose, we adopted a gold standard of 3.6K triples[1] used in the original evaluation of SCICERO to simulate various workflows incorporating HiL and LLM-iL validation modules. We designed these workflows to optimize precision, recall, and F1 score using a subset (600 triples) of the gold standard.

Specifically, we begin by implementing and investigating two approaches integrating a HiL module within SCICERO. We then replicate these pipelines, replacing the HiL module with an LLM-iL validation, leveraging GPT-4o[2]. While the improvement with the LLM validation module was not as pronounced as with the HiL, we observed a significant increase (up to 12%) in precision without any additional manual effort. Furthermore, we explore hybrid workflows that integrate both LLM-iL and HiL modules within the SCICERO framework. The results indicate that even minimal HiL involvement can lead to higher-quality extractions. Notably, an effective method for reducing human involvement and increasing scalability involved utilizing the HiL module only to resolve disagreements between automated validators.

We assess the proposed extended SCICERO workflows by testing their performance on the full gold standard dataset, confirming that the observed score improvement trends persist.

The remainder of this paper is structured as follows. Section 2 reviews related research in this area. Section 3 provides an overview of the current SCICERO pipeline and its relevant validation modules. In Section 4, we describe the newly implemented HiL and LLM-iL validation modules. The extended SCICERO pipelines and the methods used to design them are described in Section 5, followed by a discussion of the evaluation results in Section 6 and a conclusion in Section 7.

2. Related Work

This paper is situated in the intersection of three research areas: (1) methods for generating scientific knowledge graphs, (2) expertise comparison of LLMs and HiL, and (3) quality enhancement of KGs leveraging LLMs. This section offers a brief overview of related work from each of these areas.

Scientific Knowledge Graph Curation. There are two types of curation processes of scientific knowledge graphs- manual and automated. An exemplary result of the manual curation process is the Open Research Knowledge Graph (ORKG) [2]. ORKG describes articles, their contributions, applied methods, evaluation methodologies, etc. The framework relies on researchers describing the content of their scientific work manually as RDF triples. While such curation approaches allow for good quality graphs, they require high amounts of manual work and are, therefore, limited in terms of scalability.

In contrast, automatic approaches for the generation of scientific knowledge graphs can cover a high number of articles. SCICERO- the Computer Science Knowledge Graph generation pipeline [4], is an example of such automatic curation workflow. While it involves two automatic validator modules aiming to remove noisy triples, the quality of the graph cannot be guaranteed since the extracted triples are not checked by a HiL with expertise in the domain. To overcome the lack of scalability of an additional HiL validation module, we investigate additional alternative validation modules which can extend the SCICERO pipeline.

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

[2] GPT4 Omni. https://openai.com/index/hello-gpt-4o/

LLM-in-the-Loop as an Expert-in-the-Loop. Recently, LLMs have gained much research attention and have been applied in a variety of tasks across differed domains. For instance, LLMs have been prompt to take qualification tests in non-trivial domain such as clinical chemistry [12], or logic-based domain such as ontology modeling [13] showing performance comparable to the top graduate students.

LLMs have also been compared to HiL for specific human intelligence tasks. For instance, in [] LLMs judge the quality of automatically extracted texts and produce annotations similar to experts' judgments. Motivated by the results presented in literature, we investigate whether LLMs are suited for the validation of scientific knowledge graphs.

LLMs for Semantic Resource Validation. Large language model advancements have inspired several works in the area of semantic resource (i.e, knowledge graphs, ontologies, etc.) validation. The usage of LLMs for the detection of ontology modeling errors is employed in [13] for ontology restriction defect detection and in [14] for class membership validations. In [11] the authors present an LLM-based knowledge graph generation workflow, which includes a triples validation step. However, the paper does not discuss concrete validation results. Complementary to this line of work, NeOn-GPT [15] integrates an LLM for the correction of ontology errors found through external services.

The mentioned evaluation approaches are tailored towards small resources with focus on the ontological schema of knowledge graphs. Nevertheless, they obtain promising results motivating the exploration of LLMs for more complex domains.

In this paper, we propose a variety of KG validation workflows harnessing a HiL, an LLM-iL, or both of these modules to overcome the trade-off between KG quality and scalability.

3. SCICERO: The CS-KG Generation Pipeline

In this section, we briefly introduce the CS-KG generation pipeline called SCICERO [4] with a focus on its validation modules. SCICERO takes as inputs a set of scientific texts and an ontology used to semantically describe the domain knowledge in the field. The pipeline (visualized in Fig. 1) contains three main steps: (1) *extraction*, which exploits the CSO classifier [16] and revised natural language processing modules based on the CoreNLP suite [17] to produce initial sets of triples; (2) *entity and relationship handling*, which merges similar entities, filters generic entities, maps similar relationships on the same relation label, and integrates the triples coming from the various extractors into a single set; and (3) *triple validation*, consisting of two validation modules namely a *Transformer Validator* and an *Ontology-based Validator* aiming to discard incorrectly extracted or generated triples.

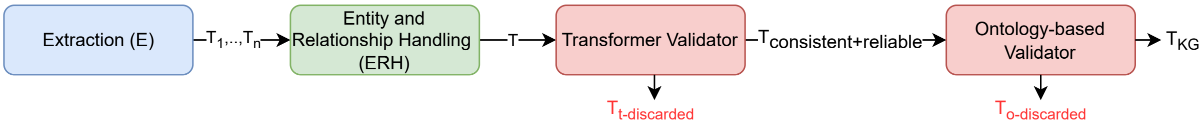

Figure 1: Simplified representation of the SCICERO [4] pipeline.

Transformer Validator. The transformer-based validation takes as input the set of refined triples produced in the extraction and subsequent entity and relationship handling step. The triples include additional information about their support-level, i.e., a number that indicates the amount of papers from which a triple was generated. The support can be interpreted as a confidence number that indicates how reliable a triple is. Intuitively, a triple that is associated with a considerable number of scientific papers (e.g., ≥ 5) is considered well-supported [4]. The transformer validator employs the following process:

(1) *define reliable and uncertain sets of triples.* $T_{relaible}$ represents highly supported statements in literature (e.g., ≥ 5), while the rest of the triples (e.g., < 5) forms $T_{uncertain}$;

(2) *fine-tune a scibert [9] transformer model* using $T_{relaible}$, and a set of negative triples $T_{negative}$ generated by corrupting $T_{relaible}$;

(3) *predict if triples from $T_{uncertain}$ are correct* and add them into the $T_{consistent}$ set or incorrect and should be discarded ($T_{t-discarded}$).

Ontology-based Validator. The ontology-based validation enables the removal of triples that do not hold according to an expert understanding of the scientific Computer Science domain. It takes as input $T_{relaible}$ and $T_{consistent}$ and filters out triples not compliant with the ontological schema. For example, a triple with a subject or object type not defined as the domain and range for the used relation will be discarded ($T_{o-discarded}$). Concretely, the triple *<dbpedia, uses, core_nlp>* would be removed since the entity *dbpedia* of type *Material* cannot use the entity *core_nlp* of type *Method*.

SCICERO Evaluation. SCICERO was evaluated using a *gold standard* of 3.6K triples *(CS-KG-3600)*. The gold standard was created by sampling the generated CS-KG and selecting 600 triples from each of the six categories: very high support triples ($\in T_{reliable}$), high support triples ($\in T_{reliable}$), low support triples ($\in T_{consistent}$), triples discarded by the transformer-validator ($\in T_{t-discarded}$), triples discarded by the ontology-validator ($\in T_{o-discarded}$), and randomly generated triples (T_{random}), which were generated by replacing the head or tail of triples from the CS-KG. Each triple was then manually annotated as correct or incorrect by 3 senior experts in the Computer Science domain. The ground truth for each triple was calculated by aggregating the annotations from the experts using a majority vote.

SCICERO has been evaluated on this set of 3.6K triples and achieves 75% precision, 79% recall and 77% F1 score. The integrated validation modules managed to significantly improve the precision from the extraction step (54% precision, 95% recall, 69% F1 score), showcasing the importance of incorporating a validation step in the extraction pipeline to filter out erroneously extracted or generated statements. Further details about SCICERO's implementation and the CS-KG can be found in [4] and [5].

4. Human-in-the-loop & LLM Validation Modules

This section describes *human-in-the-loop* and *LLM-in-the-loop* validation modules that can be attached to SCICERO to further refine the generated triples enhancing the overall quality of the CS-KG.

Human-in-the-Loop Validator. The HiL validation module is designed to incorporate expert judgments into the KG validation process. When a triple is subjected to human-in-the-loop validation, the expert judgment overwrites the automatically predicted correctness of the triple. Consequently, the triple is either incorporated into the final knowledge graph or added to the set of discarded triples $T_{h-discarded}$. In this paper, to simulate various workflows involving human participation at different stages of the validation pipeline, we utilize the available gold standard (CS-KG-3600). The HiL Validator module applies a single expert judgment, randomly chosen from the expert annotations, for each triple. We follow this approach to allow the reusability of the created gold standard while limiting biases introduced by the usage of the gold standard withing the validation pipeline.

LLM Validator. The SCICERO pipeline already includes a sciebert [9] transformer-based validation. However, the new module leverages GPT-4o for the validation of statements represented as triples. As the first step an initial prompt with instructions is sent introducing the task and expected output:

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]'.

Each following prompt includes a batch of 100 triples to be checked. Whenever the response did not follow the requested format, the batch was re-sent for validation. To reduce the variability of LLM-generated results each batch is sent 3 consecutive times and a majority vote is calculated.

5. Extended SCICERO Workflows

In this section, we present SCICERO extensions containing one or both of the new validation modules- the HiL-Validator and the LLM-Validator. To design the pipeline extensions, we explore a subset of the gold standard, used in the SCICERO evaluation as an *exploratory dataset (CS-KG-600)*. CS-KG-600 consists of 100 randomly selected triples from each of the six subsets contained within the gold standard (see Sect. 3) thus retaining a representative sample. Using CS-KG-600, we design various workflows optimizing the extraction performance. To evaluate the observed benefits of the added validation modules we test the extraction scores of the new workflows on the complete gold standard and report our findings in Section 6.

5.1. SCICERO integration with the HiL Validator

A traditional approach to improve the quality of gathered results is to involve a HiL at the end of SCICERO's existing validation pipeline (see Fig. 2). While this strategy offers an improvement in terms of the precision of the generation pipeline, the human efforts needed are enormous, especially for large resources including millions of triples such as CS-KG.

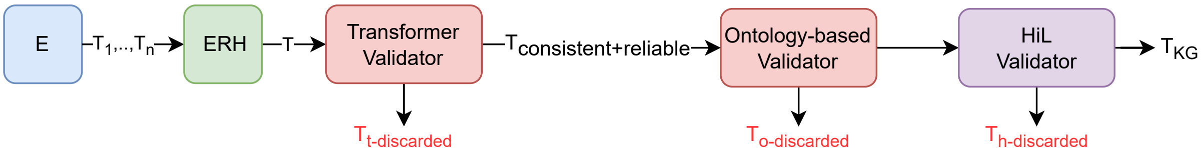

Figure 2: SCICERO integration with the HiL Validator, following a traditional HiL approach (Workflow 2).

Since the extracted triples are organized into $T_{reliable}$ and $T_{consistent}$, the solution can be adopted following the intuition that highly supported statements are correct. The modified workflow is displayed in Fig. 3. $T_{reliable}$ triples, passed through the Transformer and Ontology-based Validators are directly added to the KG. In contrast, triples with lower support ($T_{consistent}$) receive an additional HiL validation.

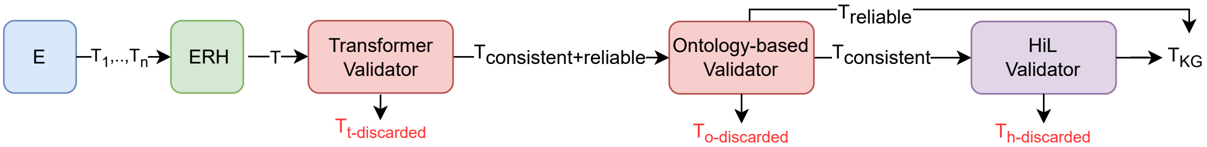

Figure 3: SCICERO integration with a HiL validation of $T_{consistent}$ triples (Workflow 3).

5.2. SCICERO integration with the LLM Validator

Related work has presented impressive results of LLMs, specifically the GPT models, for tasks typically performed by experts. We, therefore, transform the already presented workflows by replacing the HiL Validator module through the LLM Validator. Figures 4 and 5 visualize the new pipelines.

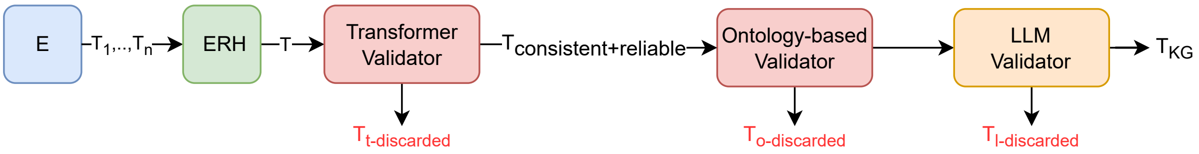

Figure 4: SCICERO integration with the LLM Validator (Workflow 4).

An objective of the replicated HiL validation is a reduction of the number of triples to be manually checked. In contrast, LLMs offer more flexibility in terms of scalable solutions. For instance, triples

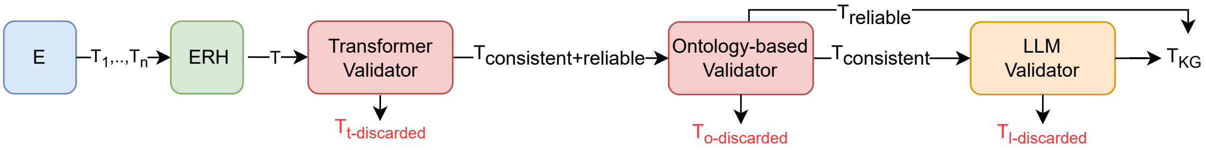

Figure 5: SCICERO integration with an LLM validation of $T_{consistent}$ triples (Workflow 5).

discarded by the Transformer Validator can be double-checked to ensure no correct triples are being removed and thus improve recall (Fig. 6).

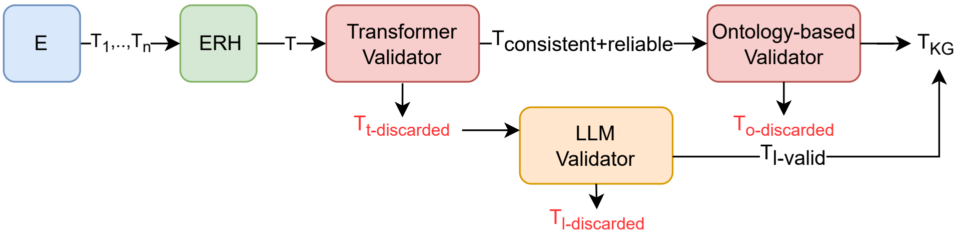

Figure 6: SCICERO integration with an LLM validation of $T_{t-discarded}$ triples (Workflow 6).

Another possibility is the inclusion of the LLM Validator at several positions within the same workflow. For instance, as shown in Fig. 7, the module can be added first, at an early stage to "rescue" discarded triples and at a later stage with the aim of removing noisy triples which may have been missed by the previous validation modules.

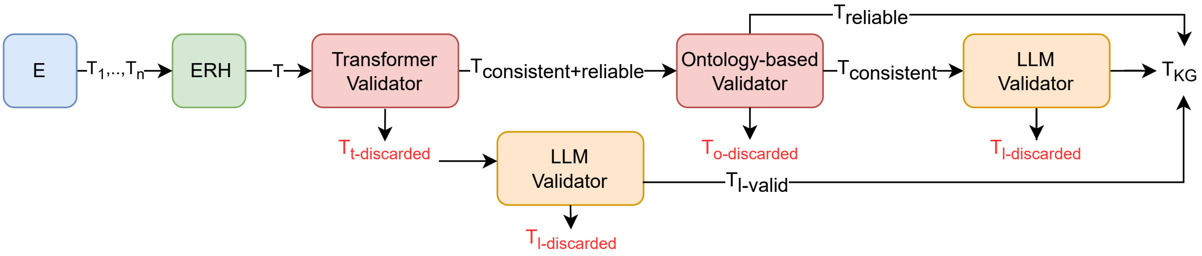

Figure 7: SCICERO integration with an LLM validation of $T_{t-discarded}$ and $T_{consistent}$ (Workflow 7).

5.3. SCICERO integration with the HiL Validator and the LLM Validator

While an LLM-iL validation can offer additional improvements to the pipeline and does not require any manual effort, it is important to keep at least some level of human oversight. Thus, we propose two exemplary hybrid workflows which can take advantage of human intelligence at scale.

Both approaches follow the notion of agreement among the automated validation modules- the original SCICERO Transformer Validator and the new LLM Validator. We follow the intuition that if multiple automated approaches assert that a triple is correct (or incorrect) it is likely to be accurate. In contrast, if a decision conflict between the modules arises, a HiL can be involved in its resolution.

Figure 8 presents a workflow following the agreement rationale for discarding triples, thus allowing the improvement of the SCICERO extraction in terms of recall, lost through the Transformer Validator.

We further extend this workflow (Fig. 9) by applying a disagreement-strategy at the final step for the remaining $T_{consistent}$ triples to ensure uncertain triples are re-evaluated before being added to the KG.

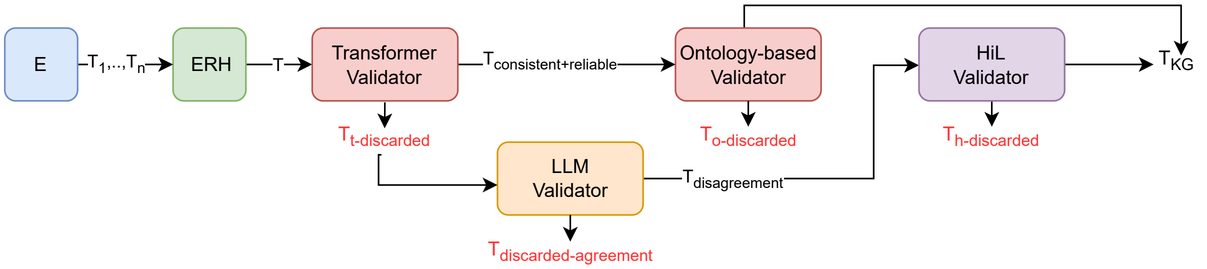

Figure 8: SCICERO integration with both the LLM and HiL validation modules, following an agreement strategy among automated validation modules for the removal of triples (Workflow 8).

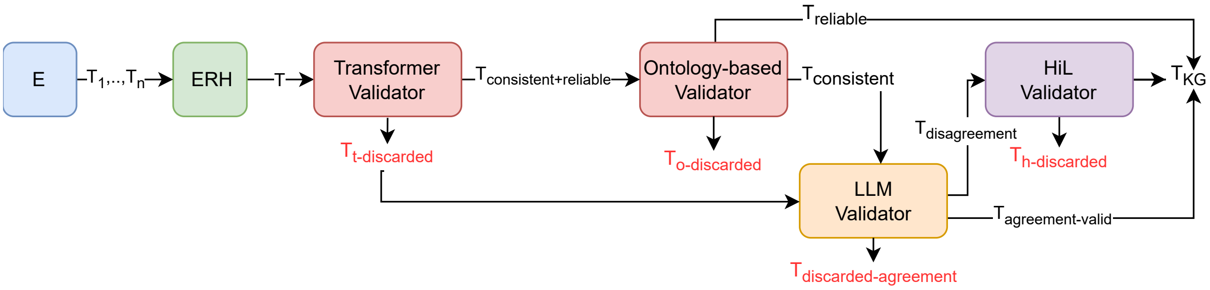

Figure 9: SCICERO integration with both the LLM and HiL validation modules, following an agreement strategy among automated validation modules for the removal of triples and need of HiL Validator (Workflow 9).

6. Results

We present the performance of the extended pipelines on the exploratory triple set and complete gold standard in Table 1. The scores are color-coded such that improvements are marked in green and a decrease is visualized in red. For each workflows discussed in Sect. 5, we include the precision, recall, F1 scores. Additionally, we provide inputs on the added validation effort in terms of number of triples to be validated either by the LLM Validator or HiL Validator.

Workflow Design Methodology. We examine whether the performance trends observed on the triple subset CS-KG-600 remain prominent as the number of triples increases. We report that for all selected workflows, as shown in Table 1, the observed improvements remain or even increase when tested against the full gold standard. These results indicate that the designed workflows are robust and scalable. Therefore, a similar exploration strategy can be employed to develop a suitable validation workflow for other automatically generated knowledge graphs, having a partial gold standard.

SCICERO Workflows Performance. To select the best-fitted validation pipeline we discuss the achieved performance with each workflow. Extensions of SCICERO with a HiL module offer improvements in terms of precision (+6% to +20%) and F1 scores (+1% to +5%). Nevertheless, significant improvements (i.e., workflow 2) require high manual efforts, which introduces a scalability issue.

As an alternative, workflows 4-7 leverage an LLM rather than a HiL. Depending on the positioning of the LLM-module either the recall or precision score can be boosted. For instance, workflow 4 reaches a precision of 85% (+12% from the baseline) on the CS-KG-600 dataset. However, a significant drop in recall (-18%) and F1 (-5%) scores is observed. In contrast, workflow 6 increases the recall to over 80% (+5%) with some losses (-2%) in the precision.

The best performing workflows, which lead to improvements across all scores for both dataset are workflows 8 and 9, employing both the LLM-Validator and HiL Validator modules. We see that the recall can be improved by 5% with minimal HiL effort (validation of approx. 5-6% of the total triples, workflow 8) without any precision losses. Similarly, workflow 9 offers improvements of both the recall (+3%) and precision (+6%) with slightly higher manual efforts (approx 12-13%).

Table 1
Results of the simulated SCICERO workflows on the CS-KG-600 exploratory dataset and the complete gold standard CS-KG-3600. For each workflow the precision (P), recall (R) and F1 scores are included as well as the added validation effort in terms of number of triples to be checked by the added LLM and HiL modules.

Workflow		**CS-KG-600**					**CS-KG-3600**				
ID	**Fig.**	**P**	**R**	**F1**	**LLM effort**	**HiL effort**	**P**	**R**	**F1**	**LLM effort**	**HiL effort**
SCICERO original pipeline (baseline)											
1	Fig. 1	73%	77%	75%	-	-	75%	79%	77%	-	-
SCICERO + HiL Validator											
2	Fig. 2	**93%**	69%	**80%**	-	300	**93%**	71%	**81%**	-	1.8K
3	Fig. 3	79%	73%	76%	-	100	**83%**	77%	79%	-	600
SCICERO + LLM Validator											
4	Fig. 4	**85%**	59%	70%	300	-	**87%**	62%	72%	1.8K	-
5	Fig. 5	78%	70%	74%	100	-	**80%**	72%	76%	600	-
6	Fig. 6	71%	**83%**	77%	100	-	73%	**84%**	78%	600	-
7	Fig. 7	76%	76%	76%	200	-	77%	77%	77%	1.2K	-
SCICERO + HiL Validator + LLM Validator											
8	Fig. 8	74%	**83%**	78%	100	**33**	76%	**84%**	79%	600	**180**
9	Fig. 9	79%	**80%**	**80%**	200	**79**	**80%**	**82%**	**81%**	1.2K	**449**

The most appropriate workflow should be selected based on the available resources and main validation goal. For instance, whenever an expert is unavailable and the elimination of noisy triples is requested, workflow 4 can be followed. In contrast, for a small KG, workflow 1 would deliver the best results. Lastly, when dealing with a large knowledge graph and only limited availability of experts, a semi-automatic validation workflow such as workflow 9 can be followed to achieve high-quality results.

7. Conclusion

In this paper, we present possible solutions extending SCICERO, the generation pipeline of the Computer Science Knowledge Graph, initially presented in [4]. We propose two new validation modules that can be integrated into the framework- one incorporating a human-in-the-loop and another LLM-in-the-loop validation leveraging GPT-4o.

Using a subset of the available gold standard from the original SCICERO evaluation as an exploratory dataset we design workflows with one or both additional validation modules optimizing the extraction scores. To evaluate the effectiveness of the proposed solutions we measured precision, recall and F1 scores on the complete gold standard, accounting for additional (manual) validation efforts.

Our findings reveal that (1) the new LLM-based validation module can increase the extraction precision by 12% reaching 85-87% (following workflow 4; Fig. 4) without any human involvement; (2) minimal manual efforts for validating 5-6% of the produced triples can lead to significant score improvements (+5% recall from workflow 8; Fig. 8); and (3) the notion of agreement among automated approaches can effectively determine the need of human-in-the-loop validation.

Each of the proposed SCICERO extensions enhances the extraction process and depending on the available resources and objective of the KG extraction, the best-suited workflow can be selected.

A limitation of this work is the exclusive focus and evaluation on a single KG generation process. Nevertheless, the added HiL and LLM-iL validation modules are developed independently from SCICERO, making the designed workflows easily adoptable to other KG generation pipelines.

In future work, we plan to employ the designed workflows in a dynamic environment, for a new subset of the CS-KG to further validate the obtained results. We further intend to use the extended SCICERO pipelines to generate various versions of the CS-KG, enabling cross-validation and comprehensive analysis of KG-enabled tasks such as hypothesis generation, forecasting of research dynamics, etc.

Acknowledgments

This work was supported by the FWF HOnEst project (V 745) and the PERKS project (101120323) 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.

References

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

[2] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, S. Auer, Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge, in: Proceedings of the 10th International Conference on Knowledge Capture, K-CAP '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 243–246. doi:10.1145/3360901.3364435.

[3] P. Groth, A. Gibson, J. Velterop, The anatomy of a nanopublication, Information services & use 30 (2010) 51–56. doi:10.3233/ISU-2010-0613.

[4] D. Dessí, F. Osborne, D. R. Recupero, D. Buscaldi, E. Motta, Scicero: A deep learning and nlp approach for generating scientific knowledge graphs in the computer science domain, Knowledge-Based Systems 258 (2022). doi:10.1016/j.knosys.2022.109945.

[5] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, Cs-kg: A large-scale knowledge graph of research entities and claims in computer science, in: U. Sattler, A. Hogan, M. Keet, V. Presutti, J. P. A. Almeida, H. Takeda, P. Monnin, G. Pirrò, C. d'Amato (Eds.), The Semantic Web – ISWC 2022, Springer International Publishing, Cham, 2022, pp. 678–696. doi:10.1007/978-3-031-19433-7_39.

[6] D. Dessì, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, H. Sack, Ai-kg: an automatically generated knowledge graph of artificial intelligence, in: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19, Springer, 2020, pp. 127–143. doi:10.1007/978-3-030-62466-8_9.

[7] A. Sakor, S. Jozashoori, E. Niazmand, A. Rivas, K. Bougiatiotis, F. Aisopos, E. Iglesias, P. D. Rohde, T. Padiya, A. Krithara, et al., 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 (2023) 100760. doi:10.1016/j.websem.2022.100760.

[8] Q. Wang, M. Li, X. Wang, N. Parulian, G. Han, J. Ma, J. Tu, Y. Lin, R. H. Zhang, W. Liu, A. Chauhan, Y. Guan, B. Li, R. Li, X. Song, Y. Fung, H. Ji, J. Han, S.-F. Chang, J. Pustejovsky, J. Rah, D. Liem, A. ELsayed, M. Palmer, C. Voss, C. Schneider, B. Onyshkevych, COVID-19 literature knowledge graph construction and drug repurposing report generation, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, Association for Computational Linguistics, 2021, pp. 66–77. doi:10.18653/v1/2021.naacl-demos.8.

[9] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. doi:10.18653/v1/D19-1371.

[10] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic Web 8 (2016) 489–508. doi:10.3233/SW-160218.

[11] H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, S. Groppe, Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text, 2023. arXiv:2305.08804.

[12] M. Sallam, K. Al-Salahat, H. Eid, J. Egger, B. Puladi, Human versus artificial intelligence: Chatgpt-4

outperforming bing, bard, chatgpt-3.5, and humans in clinical chemistry multiple-choice questions, medRxiv (2024). doi:10.1101/2024.01.08.24300995.

[13] S. Tsaneva, S. Vasic, M. Sabou, Llm-driven ontology evaluation: Verifying ontology restrictions with chatgpt, in: The Semantic Web: ESWC Satellite Events, 2024, 2024.

[14] P. T. G. Bradley P. Allen, Evaluating class membership relations in knowledge graphs using large language models, in: The Semantic Web: ESWC Satellite Events, 2024, 2024.

[15] N. Fathallah, A. Das, S. De Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: A large language model-powered pipeline for ontology learning, in: The Semantic Web: ESWC Satellite Events, 2024, 2024.

[16] A. A. Salatino, F. Osborne, E. Motta, CSO classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics, Int. J. Digit. Libr. 23 (2022) 91–110. doi:10.1007/S00799-021-00305-Y.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations, The Association for Computer Linguistics, 2014, pp. 55–60. doi:10.3115/V1/P14-5010.