# ASK-DBLP: Answering Questions over DBLP*

Tilahun Abedissa Taffa[1,2,*], Patrick Neises[4], Stefan Ollinger[4], Patrick Westphal[3], Marcel R. Ackermann[4], Debayan Banerjee[1] and Ricardo Usbeck[1]

[1]Artificial Intelligence and Explainability, Leuphana Universität Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany
[2]Department of Informatics, University of Hamburg, Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
[3]Hamburger Informatik Technologie-Center HITEC e.V., Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
[4]Schloss Dagstuhl LZI, dblp computer science bibliography, Trier, Germany

**Abstract**

DBLP is currently serving as a source of structured information for the computer science community. Among the offered services, DBLP provides users with a SPARQL endpoint interface, enabling them to write and execute SPARQL queries on the DBLP Knowledge Graph (KG). However, not every user is familiar with the SPARQL syntax and the KG schema. Having an automated method, such as semantic parsing-based KG Question Answering (KGQA), bridges the user's SPARQL familiarity gap, where KGQA converts natural language questions into structured queries to retrieve relevant data from the KG. Nevertheless, existing KGQA systems over DBLP are not robust enough to reflect the recent changes in the DBLP schema. Hence, we propose ASK-DBLP, which accepts natural language questions, converts them to SPARQL, and provides answers. In case of unclear questions, ASK-DBLP advises users to reformulate their questions. Also, it empowers users to select their preferred correct entities among the candidate linked entities and update the SPARQL. The user can also modify the resulting SPARQL query. Finally, if the user confirms the correctness of the SPARQL query and the answer, ASK-DBLP updates the training set to further improve SPARQL generation. ASK-DBLP achieves a competitive performance over the DBLP-QuAD benchmark. The current deployed version of ASK-DBLP is available at https://ask-dblp.nliwod.org.

**Keywords**

DBLP, Knowledge Graph, Knowledge Graph Question Answering, Question Answering, Question to SPARQL

## 1. Introduction

Knowledge Graphs (KGs) have emerged as a powerful means to represent entities and their rich interrelations in a structured and machine-actionable way [1]. A KG organizes data into entities and relationships, capturing diverse and complex real-world knowledge in a structured representation. By embedding semantics directly into the connections between concepts, the structured data representation paradigm enhances data integration and enables more advanced forms of automated reasoning and data discovery [1]. Scholarly KGs, like DBLP KG [2], ORKG [3], and SemOpenAlex [4], connect different scholarly entities, providing a semantic basis for bibliographic systems.

However, interaction with KGs often requires expertise in formal query languages, such as SPARQL, making access challenging for non-experts. Even KG experts cannot master every detail of a new KG; they need to understand its schema to query it effectively. Having a Question Answering[1] (QA) system on top of the KG also benefits experts by enabling them to quickly explore and comprehend the schema and content of the KG without needing to inspect its structure manually [5]. Hence, we develop ASK-DBLP, a KGQA system on top of DBLP, enabling users to interact intuitively with and obtain facts from the richly connected data. Unlike standard KGQA systems that take a question and provide an answer [6, 7], ASK-DBLP provides a SPARQL query, allowing the user to edit or

*ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan*

*Corresponding author.
tilahun.taffa@uni-hamburg.de (T. A. Taffa); patrick.neises@dagstuhl.de (P. Neises); stefan.ollinger@dagstuhl.de (S. Ollinger); patrick.westphal@uni-hamburg.de (P. Westphal); marcel.r.ackermann@dagstuhl.de (M. R. Ackermann); debayan.banerjee@leuphana.de (D. Banerjee); ricardo.usbeck@leuphana.de (R. Usbeck)
https://www.leuphana.de/en/institutes/iis/persons/ricardo-usbeck.html (R. Usbeck)
0000-0002-2476-8335 (T. A. Taffa); 0000-0002-3419-2544 (P. Neises); 0000-0001-6548-5190 (S. Ollinger); 0000-0002-3855-4485 (P. Westphal); 0000-0001-7644-2495 (M. R. Ackermann); 0000-0001-7626-8888 (D. Banerjee); 0000-0002-0191-7211 (R. Usbeck)



[1]Question Answering is the task of answering questions over a certain data source.

trigger regeneration by altering the linked entities. All resources related to ASK-DBLP are found at https://github.com/semantic-systems/ask-dblp.

## 2. Related Works

The first KGQA benchmark developed over DBLP is DBLP-QuAD [8], which introduced a dataset and a baseline model. DBLP-QuAD consists of 10K templated question-query-answer pairs that cover a range of complexities. To assess the models' generalization ability, specific template tuples and question templates are excluded from the training set but are present in the validation and test sets. As a baseline, DBLP-QuAD employs a fine-tuned T5 model (both T5-Base and T5-Small versions), following the approach in [9]. The T5 model is trained to generate SPARQL queries, taking the concatenation of the question with entity and relation URIs as input.

NLQxform [10] leverages the BART[2] model to convert questions into logical forms that structurally resemble SPARQL queries but retain entity mentions instead of URIs. Subsequently, entity mentions are resolved to their corresponding DBLP URIs using the KG search APIs. The resulting logical forms are then refined with templates and executed against the SPARQL endpoint to retrieve answers. More recently, NLQxform-UI [11] introduced a human-in-the-loop interface for QA over DBLP, while maintaining the same underlying methodology as NLQxform. BERTOlogyNavigator [12] first retrieves one-hop neighboring entities of the mentioned entity, then ranks candidate pairs by calculating the cosine similarity between the question and the pairs using BERT. The most relevant pairs are selected and filtered through heuristic rules to provide the final answer.

Despite their merits, these approaches have certain limitations: the DBLP-QuAD baseline requires extensive training data and is highly dependent on the availability of entity and relation URIs. Both NLQxform and BERTOlogyNavigator suffer from limited robustness, as they heavily rely on training sets and SPARQL templates. In contrast, our approach offers greater robustness by guiding SPARQL query generation with the DBLP schema and drawing on similar question-SPARQL pairs from the training set. Furthermore, our system has a self-learning feature, incrementally expanding the training data by incorporating newly generated and user-verified question-SPARQL pairs. This schema-driven query generation significantly enhances robustness compared to previous methods, including those that utilize Large Language Models (LLMs) [6].

## 3. Method

ASK-DBLP follows the steps shown in Figure 1. The following sub-sections describe individual components.

### 3.1. Question Clarity Checking

ASK-DBLP first assesses the completeness and clarity of a user's question. This is achieved by requesting Qwen 2.5[3] hosted by Chat-AI [13] using the prompt found in the ASK-DBLP additional resources wiki page of this paper[4]. If a question is vague or incomplete, the system politely prompts the user to revise the question, providing guidance for clarification. For example, ambiguous questions like "Give me the best database paper" are flagged for revision because it is unclear whether the user refers to a best-paper award or a citation impact. This initial phase helps refine user input and ensures that subsequent steps operate on well-formed questions.

[2] facebook/bart-base, fine-tuned on question-SPARQL pairs from DBLP-QuAD. SPARQL syntax elements (such as SELECT, COUNT, ORDER BY), parentheses, and DBLP-specific relations are added as special tokens to improve the logical form conversion.
[3] https://qwenlm.github.io/blog/qwen2.5-coder-family/
[4] https://github.com/semantic-systems/ask-dblp/wiki/ASK%E2%80%90DBLP-ISWC-2025-Publication-Additional-Resources

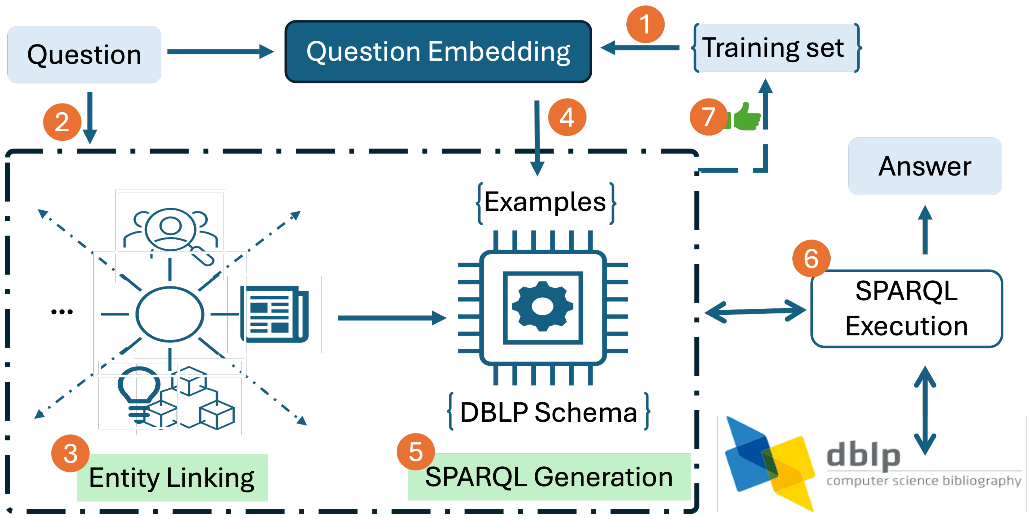


**Figure 1:** ASK-DBLP Model. ASK-DBLP operates as follows: (1) Embeds training questions offline. When a user enters a question, (2) checks question clarity, and (3) the entity linker identifies relevant entities within the question. At the same time, (4) the system retrieves the top-5 most similar questions from the training set, along with their associated SPARQL queries, by comparing the user question embedding to the training embeddings. (5) The SPARQL generator prompts the LLM and generates a SPARQL query. (6) Executes the generated SPARQL and returns the resulting answer to the user. (7) If the user responds with thumbs up for the SPARQL query and the answer, the training set is updated accordingly.

### 3.2. Entity Linking and Retrieval of Similar Questions

The user question is then passed to an entity linker module [14, 15], which extracts candidate entities and maps them to their respective URIs in DBLP. If the entity linker fails to identify relevant entities, it alerts the SPARQL generator, signaling that no entities are recognized in the input question. Following that, ASK-DBLP identifies the top 5 similar questions from the training set, following the approach described in [6].

### 3.3. SPARQL Generation

The SPARQL generator constructs a prompt that includes the question, the retrieved examples, the linked entities, and the DBLP schema. The SPARQL generation prompt (see the ASK-DBLP additional resources wiki page of this paper[5]) instructs the Qwen 2.5 LLM to generate an appropriate SPARQL query for the given user question. If no entities are identified in the question, the prompt is modified to include only the schema and the example questions. When similar questions cannot be determined, the LLM receives the question and the schema for SPARQL generation. Upon successful SPARQL generation, if the query produces valid results and the user provides "thumbs up" feedback, the pair comprising the user question and the generated query is added to the training set, enabling the system to improve incrementally through new examples.

Moreover, the system allows users to edit the generated SPARQL queries or select different entities from the entity linker results, and then request a regeneration of the query as needed. This flexibility supports iterative improvements and better aligns the query to user intent. Finally, when the user chooses to run the SPARQL query, it is executed against DBLP's SPARQL endpoint. As shown in Figure 2, the resulting answers are presented to the user in tabular format.

[5]https://github.com/semantic-systems/ask-dblp/wiki/ASK%E2%80%90DBLP-ISWC-2025-Publication-Additional-Resources


**Figure 2:** ASK-DBLP User Interface.

## 3.4. ASK-DBLP User Interface

The user interface[6] is shown in Figure 2, with different sections labeled by numbers. At first, the user is prompted to enter a natural language question in the input field (1). For inspiration, a couple of example questions are also displayed. Once the question is formulated, the "Generate SPARQL"-button (2) can trigger the generation process. Once this is completed, the user is presented with a selection of entities and the candidates linked to them by the entity linker. Various candidates are presented for each entity, and the user can select the correct entity using the dropdown menus. When the selection process is finished, the query can be updated with the selected values by clicking the "Update SPARQL

[6]ASK-DBLP UI is built with Next.js (https://nextjs.org/docs) and seamlessly integrated with a backend implemented entirely in Python using the Flask framework.

with Selected Entities"-button (3). In (4), the obtained query is presented and can be modified by the user, if needed. By pressing the "Run Query"-button (5), the query is evaluated by using the public DBLP SPARQL endpoint[7] and the results are displayed (6). When the user is satisfied with the query and the results, it is optional to use the "thumbs up"-button (7) to provide the question and the SPARQL query for further training as mentioned in Section 3.3.

## 4. Evaluation

The evaluation of ASK-DBLP employs DBLP-QuAD[8], a dataset created from an earlier DBLP dump[9]. The experimental result of our model is compared to the existing method's performance in Table 1. Our approach achieves an F1 score of 0.7614, outperforming the DBLP-QuAD baseline (T5 Small), Jiang et al. [16], BERTologyNavigator [12], and PSYCHIC [17]. Although the DBLP-QuAD baseline (T5 Base) achieves the highest F1 score of 0.868, our model demonstrates strong performance given its relative simplicity and efficiency. The strength of NLQxform lies in its in-house developed entity linker. In contrast, our approach utilizes an openly available entity linking tool. The DBLP-QuAD baseline operates in a pre-linked setting, which contributed to higher performance, using pre-identified entities and relations during the SPARQL generation.

| Method | F1 - Score |
|---|---|
| DBLP-QuAD baseline (T5 Base) [8] | **0.8680** |
| NLQxform [10] | 0.8488 |
| DBLP-QuAD baseline (T5 Small) [8] | 0.7210 |
| Jiang et al. [16] | 0.6619 |
| BERTologyNavigator [12] | 0.2175 |
| PSYCHIC [17] | 0.1800 |
| ASK-DBLP (Ours) | <u>0.7614</u> |

**Table 1**
F1 scores comparison of ASK-DBLP and other KG QA systems over the DBLP-QuAD.

Unlike the DBLP-QuAD T5 Base model, our method requires only a minimal training set, utilizing similar questions as inline context during SPARQL generation. Moreover, we do not train any language model from scratch; instead, we leverage a publicly available model, specifically Qwen 2.5 Coder[10] In contrast to NLQxform, which relies heavily on SPARQL templates derived from the DBLP-QuAD dataset before SPARQL generation, ASK-DBLP generates SPARQL queries without using any predefined templates. Instead, it is guided by automatically selected examples and the DBLP schema. Alternative methods necessitate retraining and template updates from scratch to adhere to the continuous evolution of the DBLP KG. In comparison, ASK-DBLP is more robust, as it dynamically follows the schema and generates SPARQL queries driven by a small set of example queries. Furthermore, ASK-DBLP incorporates a self-improving feature by leveraging user feedback. Incorporating user-verified and voluntarily submitted questions and queries enables continual improvements and adaptation.

[7] https://sparql.dblp.org/
[8] DBLP-QuAD is a well-established KGQA benchmark tailored to the computer science domain, providing a reliable and widely used basis for evaluating KGQA methods. Using DBLP-QuAD enables direct comparison with existing approaches, facilitating objective assessment and tracking progress within the research community.
[9] https://blog.dblp.org/2022/03/02/dblp-in-rdf/
[10] Since our goal was not to comprehensively evaluate LLMs but to select one that best suited our task, we performed a manual random evaluation of a few open-source models and selected Qwen 2.5 Coder. Its familiarity with structured query languages such as SQL enhances its ability to produce well-structured SPARQL queries. Moreover, Qwen 2.5 Coder follows instructions reliably and, for our specific task, makes it a highly suitable choice.

## 5. Conclusion, Limitations, and Future Work

We propose ASK-DBLP, a KGQA system over DBLP, that leverages the KG schema and a few examples during SPARQL generation, making it easier to adapt to the continuously evolving DBLP KG. Moreover, our proposed method enables the collection of self-checked question-SPARQL pairs from users. Those new examples are used to improve the SPARQL generation and can be shared with the research community.

One limitation is that we are currently using an entity linker and LLM APIs; we plan to decouple from the high API usage calls by deploying them locally and further reducing latencies. As one direction of future work, we will explore different entity linking methods. We currently use few-shot in-context learning-based SPARQL generation over an open-source general purpose LLM; fine-tuning the LLM for SPARQL generation is another future direction.

## Acknowledgments

This work is supported by the German Research Foundation (DFG) consortium NFDI4DataScience under Grant No. 460234259 and consortium NFDIxCS Grant No. 501930651. We utilized two NVIDIA RTX A5000 24GB GPUs, kindly provided by the NVIDIA Academic Hardware Grant Program.

## Declaration on Generative AI

While preparing this work, the author(s) used Grammarly and DeepL to: Grammar and spelling check. Further, the author(s) used the Flux[11] model hosted by Chat-AI to: Generate images (the ASK-DBLP logo shown on top of Figure 2). After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3447772. doi:10.1145/3447772.

[2] M. R. Ackermann, H. Bast, B. M. Beckermann, J. Kalbach, P. Neises, S. Ollinger, The dblp Knowledge Graph and SPARQL Endpoint, TGDK 2 (2024) 3:1–3:23. URL: https://doi.org/10.4230/TGDK.2.2.3. doi:10.4230/TGDK.2.2.3.

[3] 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: M. Kejriwal, P. A. Szekely, R. Troncy (Eds.), Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019, ACM, 2019, pp. 243–246. URL: https://doi.org/10.1145/3360901.3364435. doi:10.1145/3360901.3364435.

[4] M. Färber, D. Lamprecht, J. Krause, L. Aung, P. Haase, SemOpenAlex: The Scientific Landscape in 26 Billion RDF Triples, in: International Semantic Web Conference, Springer, 2023, pp. 94–112.

[5] R. Omar, I. Dhall, P. Kalnis, E. Mansour, A Universal Question-Answering Platform for Knowledge Graphs, Proc. ACM Manag. Data 1 (2023) 57:1–57:25. URL: https://doi.org/10.1145/3588911. doi:10.1145/3588911.

[6] T. A. Taffa, R. Usbeck, Leveraging LLMs in Scholarly Knowledge Graph Question Answering, in: D. Banerjee, R. Usbeck, N. Mihindukulasooriya, G. Singh, R. Mutharaju, P. Kapanipathi (Eds.), Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023, volume 3592 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper5.pdf.

[11] https://flux1.ai

[7] A. Perevalov, X. Yan, L. Kovriguina, L. Jiang, A. Both, R. Usbeck, Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 2998–3007. URL: https://aclanthology.org/2022.lrec-1.321/.

[8] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, DBLP-QuAD: A Question Answering Dataset over the DBLP Scholarly Knowledge Graph, in: I. Frommholz, P. Mayr, G. Cabanac, S. Verberne, J. Brennan (Eds.), Proceedings of the 13th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 45th European Conference on Information Retrieval (ECIR 2023), Dublin, Ireland, April 2nd, 2023, volume 3617 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 37–51. URL: https://ceur-ws.org/Vol-3617/paper-05.pdf.

[9] D. Banerjee, Semantic Parsing for Knowledge Graph Question Answering with Large Language Models, in: C. Pesquita, H. Skaf-Molli, V. Efthymiou, S. Kirrane, A. Ngonga, D. Collarana, R. Cerqueira, M. Alam, C. Trojahn, S. Hertling (Eds.), The Semantic Web: ESWC 2023 Satellite Events - Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings, volume 13998 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 234–243. URL: https://doi.org/10.1007/978-3-031-43458-7_42. doi:10.1007/978-3-031-43458-7_42.

[10] R. Wang, Z. Zhang, L. Rossetto, F. Ruosch, A. Bernstein, NLQxform: A Language Model-based Question to SPARQL Transformer, in: D. Banerjee, R. Usbeck, N. Mihindukulasooriya, G. Singh, R. Mutharaju, P. Kapanipathi (Eds.), Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023, volume 3592 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper2.pdf.

[11] R. Wang, Z. Zhang, L. Rossetto, F. Ruosch, A. Bernstein, NLQxform-UI: An Interactive and Intuitive Scholarly Question Answering System, in: N. Ferro, M. Maistro, G. Pasi, O. Alonso, A. Trotman, S. Verberne (Eds.), Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025, ACM, 2025, pp. 3990–3993. URL: https://doi.org/10.1145/3726302.3730153. doi:10.1145/3726302.3730153.

[12] S. Rajpal, R. Usbeck, BERTologyNavigator: Advanced Question Answering with BERT-based Semantics, CoRR abs/2401.09553 (2024). URL: https://doi.org/10.48550/arXiv.2401.09553. doi:10.48550/ARXIV.2401.09553. arXiv:2401.09553.

[13] A. Doosthosseini, J. Decker, H. Nolte, J. M. Kunkel, Chat AI: A Seamless Slurm-Native Solution for HPC-Based Services, 2024. URL: https://arxiv.org/abs/2407.00110. arXiv:2407.00110.

[14] D. Banerjee, Arefa, R. Usbeck, C. Biemann, DBLPLink: An Entity Linker for the DBLP Scholarly Knowledge Graph, in: I. Fundulaki, K. Kozaki, D. Garijo, J. M. Gómez-Pérez (Eds.), Proceedings of the ISWC 2023 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 6-10, 2023, volume 3632 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3632/ISWC2023_paper_428.pdf.

[15] Banerjee, Debayan and Taffa, Tilahun Abedissa and Usbeck, Ricardo, DBLPLink 2.0 - An Entity Linker for the DBLP Scholarly Knowledge Graph, arXiv preprint https://arxiv.org/abs/2507.22811 (2025).

[16] L. Jiang, X. Yan, R. Usbeck, A Structure and Content Prompt-based Method for Knowledge Graph Question Answering over Scholarly Data., 2023.

[17] H. A. Akl, PSYCHIC: A Neuro-Symbolic Framework for Knowledge Graph Question-Answering Grounding, in: D. Banerjee, R. Usbeck, N. Mihindukulasooriya, G. Singh, R. Mutharaju, P. Kapanipathi (Eds.), Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023, volume 3592 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3592/paper1.pdf.