Integrating I4.0 Knowledge Graphs with Large Language Models Beyond SPARQL Endpoints

Abdul Wahid^{1,*,†}, Muhammad Yahya^{2,†}, Farooq Zaman^{3,†}, Baifan Zhou⁴, John G. Breslin⁵, Muhammad Ali Intizar⁶ and Evgeny Kharlamov⁷

¹Data Science Institute, University of Galway, IDA Business Park, Lower Dangan Galway, Ireland H91 AEX4

³Valeo Vision Systems, Tuam, Galway, Ireland

³Information Technology University, Lahore, Pakistan

⁴Department of Computer Science, Oslo Metropolitan University, Oslo, Norway

⁵Data Science Institute, University of Galway, IDA Business Park, Lower Dangan Galway, Ireland H91 AEX4

⁶School of Electronic Engineering, Dublin City University, Dublin, Ireland

⁷Bosch Center for AI, Renningen, Germany

Abstract

Industry 4.0 (I4.0) knowledge graphs are a common way to represent industrial information models. Conventional SPARQL querying systems require the users to be familiar with the data schema and SPARQL syntax. However, this is often very difficult for many users in industrial production, who have mostly an engineering background, instead of a semantic web. Recent developments in large language models (LLMs) make it possible for non-semantic experts to use natural language to query knowledge graphs (KG). In this work, we present a framework and preliminary results of integrating Industry 4.0 KGs with LLMs to improve how data is represented, reasoned, and processed in manufacturing contexts, facilitating user interaction with KGs and contributing to operational efficiency. Our technique enhances Language Models (LLMs) by utilising the semantic complexity and interdependence of Knowledge Graphs (KGs). This allows us to incorporate domain-specific knowledge. We used the FAISS library and LLaMA2 to optimise the storage and retrieval of vectors, which improved the system's performance and scalability. This integration allows for advanced fault detection, proactive maintenance, and process optimisation, resulting in decreased periods of inactivity and improved productivity. We introduce the framework's architecture, implementation strategy, and possible advantages while also discussing the difficulties associated with data integration and scalability. The results of our study show that the integration of KG-LLM surpasses traditional approaches in terms of operational efficiency, as evidenced by enhanced fault detection, proactive maintenance, and process optimisation, thereby opening up possibilities for the advancement of more intelligent and resilient production systems.

Keywords

Knowledge Graphs, LLaMa, SPARQL, Large Language Model (LLM), Deep Learning

1. Introduction

In Industry 4.0 (I4.0), digital twins of industrial assets are built thanks to the massive collection and representation of industrial data and knowledge. Industrial information models are a widely applied type of digital twin, and their adoption continues to draw increasing attention. Knowledge Graphs (KG) are a common way to represent industrial information models [1]. A Knowledge Graph is a data

SemIIM2024 (Third International Workshop on Semantic Industrial Information Modelling, Nov 11, 2024 – Nov 15, 2024, Baltimore, USA

^{*}Corresponding author.

[†]These authors contributed equally.

 [▲] a.wahid2@universityofgalway.ie (A. Wahid); muhammad.yahya@valeo.com (M. Yahya); phdcs18002@itu.edu.pk
 (F. Zaman); baifan.zhou@oslomet.no (B. Zhou); john.breslin@universityofgalway.ie (J. G. Breslin);

evgeny.kharlamov@de.bosch.com (E. Kharlamov)

thtps://github.com/AbNeil (A. Wahid); https://www.oslomet.no/en/about/employee/bazho0342/ (B. Zhou); https://www.johnbreslin.com/ (J. G. Breslin); https://www.dcu.ie/electronics/people/ali-intizar (M. A. Intizar);

https://www.bosch.com/research/about-bosch-research/our-research-experts/evgeny-kharlamov/ (E. Kharlamov)

^{© 0000-0003-2625-276}X (A. Wahid); 0000-0002-5766-5172 (M. Yahya); 0000-0003-3698-0541 (B. Zhou); 0000-0001-5790-050X (J. G. Breslin); 0000-0002-0674-2131 (M. A. Intizar); 0000-0003-3247-4166 (E. Kharlamov)

^{© 02022} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

structure designed to gather and share information about the real world. The nodes in the network represent entities of interest, while the edges represent various relationships between these entities [2]. It semantically models the data in a structured way and means and extracts knowledge using deductive and inductive techniques [2]. In recent years, the use of Knowledge Graphs has become essential in various industrial companies, including Bosch [3], Siemens [4], Airbus [5], and others. KGs address the issue of diverse data by unifying the data sources into a cohesive structure, enabling streamlined data retrieval through a SPARQL endpoint [13]. This integration reduces the effort required for data access and enhances data connectivity and cohesiveness across the enterprise. Figure 1 illustrates the architecture of the enterprise system that utilises the KGs for integrated data retrieval.

To demonstrate our proposed framework's practical application and effectiveness, we utilise a specific use case: a Football Production Line. This production line consists of multiple machines responsible for various stages of football manufacturing, including material preparation, stitching, and quality control, as shown in Figure 1. Each machine generates vast amounts of data related to processes, tools, sensor readings, and operational parameters. This data is essential for real-time monitoring, fault detection, and process optimisation, but the complexity of querying such diverse information through traditional methods like SPARQL presents a significant barrier for non-expert users. Therefore, by integrating a Knowledge Graph (KG) with large language models (LLMs), we enable natural language querying, making the data more accessible to users without semantic web expertise. Figure 1 illustrates the architecture of the generalised framework we propose, which can be applied across various industrial settings, with the Football Production Line serving as our use case to validate the system.

However, to maximise the utility of these SPARQL endpoints, it is crucial to provide comprehensive training for manufacturing line end users, including engineers, supervisors, operators, and other personnel, who are normally non-semantic experts and are not familiar with technologies such as ontology and SPARQL. This lack of expertise and excessive training costs impede the adoption and development of industrial information models represented in KGs.

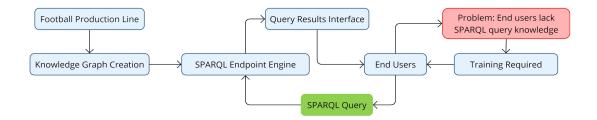
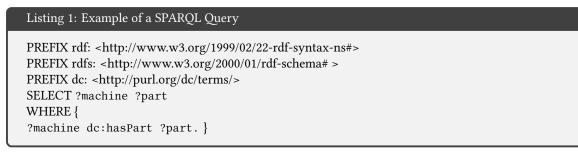


Figure 1: Overview of the enterprise system employing a Knowledge Graph for unified data retrieval.



The latest progress in Large Language Models (LLMs) provides an opportunity to support users in visually examining and analysing KGs. LLMs like BERT, RoBERTa, and T5, pre-trained on extensive corpora, excel in various NLP tasks such as question answering, machine translation, and text generation [8]. Advanced LLMs like ChatGPT and PaLM2, with billions of parameters, show great promise in complex tasks such as education, code generation, and recommendation [9]. Similarly, the end users in

the manufacturing line of I4.0 can leverage the KGs by integrating them with LLMs where they can fetch the information using natural language.

In this paper, we propose an approach that utilises LLMs for user interaction with industrial KGs, reducing the training time, cost, and expertise barrier that I4.0 users may confront within the conventional SPARQL querying approach. The paper provides a literature review. (Section 2), presents the proposed architecture (Section 3) provides experimental analysis (Section 4). Finally, section 5 concludes the paper and suggests possible future work.

2. Literature Review

2.1. Knowledge Graphs in Manufacturing

In recent years, Knowledge Graphs (KGs) have played a crucial role in creating digital twins (DT) of physical systems for Industry 4.0, to improve the management of manufacturing processes [10], [11]. Furthermore, the implementation of I4.0 has played a crucial role in resolving concerns regarding interoperability and data integration, as stated in [12], [14]. The authors presented the Bosch Industry 4.0 Knowledge Graph (BI40KG), which is a methodology that uses ontologies to integrate data sources in a Knowledge Graph. This methodology enhances interoperability and traceability within Industry 4.0 environments [15]. A separate investigation examined the utilisation of Knowledge Graph embeddings (KGE) in integrating data for monitoring the quality of automobile welding. This study emphasised the difficulties and potential benefits associated with this approach [16]. This study addressed the complex challenge of combining data to determine the diameter of welding spots and identify car bodies. Nevertheless, there is a notable obstacle in enabling end-users, including engineers, device operators, and supervisors, to efficiently utilise these Knowledge Graphs. The current method requires intensive training sessions and workshops to educate users on how to use SPARQL endpoints. It is essential to undergo this training because SPARQL endpoints do not provide sufficient assistance for natural language searches. As a result, users must possess a thorough grasp of query formulation to obtain relevant results.

2.2. Integration of KGs and LLMs

A collaborative training-free reasoning schema that combines KGs and LLMs to address challenges faced by LLMs in practical applications such as hallucinations, knowledge updating issues, and limited transparency in reasoning was proposed in [17]. The technique involves LLMs systematically exploring the KG to retrieve knowledge subgraphs that are relevant to the task. This helps guide the LLMs in combining implicit knowledge for reasoning on the subgraph. In [18] a combination of qualitative and quantitative research methodologies was employed to investigate how LLMs might aid users in the exploration and analysis of Knowledge Graphs (KGs). Specifically, it examines the use of collaborative query formulation, multi-turn discussion for discovering relationships, and the creation of on-demand visualisations. The concept of utilising universal LLMs to create KGs for extracting knowledge from complicated texts was introduced in [19]. This was followed by the process of updating domain-specific LLMs through knowledge editing, resulting in high levels of accuracy in question-answer tasks across several domains. In [20], LLMs were assessed for their ability to create and reason with KGs. This evaluation involved eight datasets that encompassed tasks such as entity, relation, and event extraction, link prediction, and question answering. A recent study examines the use of ChatGPT in performing experiments to investigate its capabilities in assisting Knowledge Graph Engineering (KGE) [21]. The paper presents findings that indicate how ChatGPT can aid in the creation and administration of KGs.

The literature indicates that LLMs are being integrated with KGs to improve their construction and logical reasoning [20], [22], [23], [24]. Notably, models like GPT-4 outperform ChatGPT in many tasks and outperform models optimised for certain reasoning and question-answering datasets [20]. This shows that LLMs can now extract and reason about knowledge in KGs, making them useful in complicated information systems.

3. Our Approach

The primary goal of this work is to integrate Industry 4.0 Knowledge Graphs (KGs) with Large Language Models (LLMs) to enable natural language querying of industrial data. This approach allows non-semantic experts such as engineers and operators to interact with the KG using simple natural language, without needing specialized knowledge of SPARQL or ontology. The expected outcome is to increase the accessibility of industrial data, leading to improved operational efficiency, fault detection, proactive maintenance, and process optimisation.

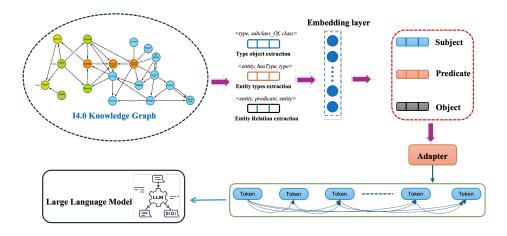


Figure 2: Integration of I4.0 KG with a large language model (LLM). The Knowledge Graph facilitates entity and relation extraction, producing the corresponding embeddings. These embeddings, representing subjects, predicates, and objects, are processed through an adapter module. The adapter integrates the structured knowledge into the LLM, enhancing its natural language understanding and generative capabilities.

3.1. Football Production Line Dataset

The football production process involves nine machines, each generating data related to tools, processes, critical parameters, and sensor outputs. This data is gathered and analysed using Reference Generic Ontology Model (RGOM) classes and relations, and subsequently mapped to ontology terms via the Jena API³, resulting in an RDF triple store referred to as the Industry 4.0 Knowledge Graph (I4.0 KG) [25]. On average, each football production cycle generates 1730 triples. With 36 footballs produced per hour, this results in approximately 22,150 triples generated hourly from 2,903 distinct individuals (entities), which represent machines, tools, sensors, and other relevant components of the production line. These triples are stored in CSV format to facilitate integration with the Large Language Model (LLM).

The RGOM serves as a set of predefined classes that categorise and organise knowledge related to the football production process. These classes represent different entities involved in production, such as machines, tools, and processes, along with their interrelations. By mapping production data to RGOM classes, we create a structured Knowledge Graph that can be queried and reasoned upon. The 1730 triples generated per football production cycle represent real-time data on various aspects of the production line, including sensor readings, machine status, tool usage, and process parameters. These triples are dynamic and continuously updated as the production process evolves. The average of 1730 triples is tied to each football's production cycle, reflecting real-time updates in the Knowledge Graph.

With 36 footballs produced each hour, the system generates approximately 22,150 triples, collected from 2,903 distinct individuals (entities such as machines, tools, and sensors). The Knowledge Graph grows dynamically with each production cycle, adding new data. Our system architecture ensures

³https://jena.apache.org

that the data is scalable, using FAISS for vector-based retrieval to efficiently handle large volumes of information and provide quick access to relevant data even as the graph grows continuously.

3.2. Overview of the Proposed Framework

A Knowledge Graph (KG) can be represented as a labelled directed graph $G = (V_e, E, T)$, where V_e and E are sets of nodes and labels representing entities and relations, respectively, and T represents the triples. Each triple (s, r, o) consists of a subject s, a relation r, and an object o. To enhance the usability and efficiency of the KG, we generate embeddings for its components and store them in a vector space.

The embedding are generated using sentence transformer, an efficient Embedding model for converting text to embeddings [26]. By leveraging sentence transformer, we can transform the KG into a vector store, where embedding for each triple are computed and stored. Specifically, for each triple (*s*, *r*, *o*), we compute embedding for the subject *s*, the relation *r*, and the object *o*. These embedding are then stored in the FAISSFAISS² vector store, enabling efficient vector-based retrieval.

To facilitate the retrieval of relevant triples from the KG, we define a retrieval function that calculates the cosine similarity between a query embedding and the pre-computed embeddings of the triples stored in the vector store by building a data structure in RAM from a given set of vectors $x_1, ..., x_n$ in dimension *d*. The cosine similarity function is computed as shown in 1. When a query is issued, the function computes the cosine similarity scores between the query embedding and all stored embeddings, retrieving the triples with the highest similarity scores. This approach allows for efficient and accurate query matching within the Knowledge Graph, leveraging the power of vector space representations and similarity search to enhance the usability of the KG for various applications.

$$i = \arg\min_{i} \| x - x_1 \| \tag{1}$$

Where $\| \cdot \|$ is the Euclidean distance (L2) and the dimension of x_i needs to be fixed.

3.3. Integration with LLM

FAISS converted our knowledge network into a vector store, and Meta AI's LLaMA2 enhanced triple explanations based on user questions. This approach required multiple phases to make information relevant and clear. First, we created and refined an LLaMA2 prompt. This prompt taught the language model to explain retrieved triples clearly and contextually. We made the prompt's explanations relevant to the user's informational needs by providing contextual information from the query. We then customised and fine-tuned LLaMA2 to comprehend our Knowledge Graph's structure and semantics. Training the model with triples and their explanations enhanced its accuracy and relevance. A user query is processed to extract relevant embeddings, and the retrieval function finds triples with the highest cosine similarity scores. After receiving these triples, fine-tuned LLaMA2 provides thorough explanations. These explanations make triple connections and entities easier to understand.

Our Knowledge Graph retrieval procedure is much more usable with LLaMA2 because it simplifies complex data. The Knowledge Graph is useful for research, education, and data-driven decision-making since the model delivers context-aware responses to user queries.

3.4. Proposed Methodology

Our strategy focused on prioritising prompt engineering rather than fine-tuning or training the LLM from scratch. Prompt engineering refers to the process of creating precise prompts to direct the LLM in producing desired results, without making changes to the model's underlying parameters or retraining it. This approach is more effective and flexible, enabling us to utilise the advantages of existing LLMs without requiring significant computational resources.

Our tests utilised an enhanced iteration of LLaMA2, developed and compiled in C++, with a specific focus on effortless integration with Python. This optimisation ensured that the model could operate

²https://ai.meta.com/tools/faiss/

query		output
what is the temperature of Machine_6 from 12: 06 12:30:31 to 06-12 12:47:57?		As a software requirements analyst and expert, I would explain the query results as follows: Triple 1: /Machine1_Temperature_Observation_716 hasSimpleResult 41 This triple indicates that at 12:06:12, Machine1 had a temperature observation of 41.
		 Triple 2: /Machine1_Temperature_Observation_727 hasSimpleResult 40 This triple indicates that at 12:30:31, Machine1 had a temperature observation of 40. Triple 3: /Machine2_Heater1_Temperature_Observation_3 hasSimpleResult 61 This triple indicates that at 12:00:03, Machine2's heater 1 had a temperature observation of 61. Triple 4: /Machine1_Temperature_Observation_1816 hasSimpleResult 41 This triple indicates that at 12:47:57, Machine1 had a temperature observation of 41. Based on the results, we can see that there are no observations of Machine_6's temperature during the time period of 12:06:12 to 06-12 12:47:57. Therefore, the output is empty.

Figure 3: The system's response to the query about the temperature of *Machine_6*. The output indicates no temperature observations within the specified period.

with greater efficiency and effectiveness within our current Python-based infrastructure. Through prioritising prompt engineering, we successfully utilised the capabilities of LLaMA2 to produce concise and contextually appropriate explanations for the triples obtained from our KG. This improved the user experience significantly without requiring considerable retraining of the model.

4. Results and Discussions

Instead of relying on traditional SPARQL queries, our approach leverages natural language prompts with the help of LLaMA2. This enables end users to interact with the Knowledge Graph in plain language, significantly enhancing accessibility and ease of use. For example, as shown in Figure 3, when queried about the temperature of *Machine*₆ from "12:06 12:30:31 to 06-12 12:47:57," the system analyzed the query and retrieved the relevant triples from the Knowledge Graph. The results indicated that no observations of Machine_6's temperature were found during the specified period. Listing 2 shows the SPARQL query that typically returns this result. Using natural language queries removes the need for SPARQL understanding or logical reasoning, making it easier for non-experts to access and query the Knowledge Graph without requiring specialized query language knowledge.

```
Listing 2: SPARQL to Query retrieve the temperature results for machine_6 for a specified time

PREFIX smo: <http://www.semanticweb.org/manufacturingproductionline/>
PREFIX d: <http://www.semanticweb.org/manufacturingproductionline/data/>
PREFIX tm: <http://www.w3.org/2006/time#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
SELECT DISTINCT ?machine ?Start_time ?result
WHERE{ ?machine smo:hasTool ?tool.
?tool sosa:madeObservation ?observation.
?observation sosa:hasSimpleResult ?result.
?time tm:hasStartTime ?Start_time. }
```

Similarly, as shown in Figure 4, when queried about the status of the *machine*₂ motor from "2021-06-01T 10:11:00Z to 2021-06-01T 10:12:55Z", the system determined that the motor was continuously working throughout the specified time range, with no changes in status. The retrieved triples, such as /Machine2_Motor_State_2021 hasState working, confirmed the motor's operational state. Like the previous example, Listing 3 shows that natural language querying is intuitive and easy, allowing non-domain specialists to retrieve the necessary information from the Knowledge Graph without needing



Figure 4: The system's response to the query about the status of the machine_2motor.

to understand SPARQL or other complex query languages.

Our proposed methodology leverages the advanced capabilities of LLaMA2 to interpret these natural language queries. Through prompt engineering, we customised the model's responses to generate detailed and contextually relevant explanations for each retrieved triple. This approach allows us to achieve the desired output without performing additional model training or fine-tuning. When user queries are turned into embeddings, the retrieval function finds the cosine similarity scores between the query embedding and the triples' embeddings that have already been computed and stored in the vector space. As a result, our method makes Knowledge Graphs easier to use by letting people interact with them using natural language and supporting it with the advanced interpretation skills of LLMs. This lets people who do not know SPARQL before have a smooth and effective querying experience.

The examples in this work utilise exact timestamps (e.g., "12:06:12 to 12:47:57") because the Knowledge Graph stores precise sensor data at specific times. However, we recognise that typical users may input broader, more natural time ranges (e.g., "12:00 to 12:30") when querying the system. Currently, the system requires an exact match with the stored timestamps to retrieve the relevant triples. This reliance on precision can pose a limitation, as queries that do not exactly match the stored times (e.g., "12:00" instead of "12:06:12") may fail to return results. To overcome this limitation, we propose enhancing the system with a time-range estimation mechanism. This enhancement would allow the system to interpret user-specified time ranges and map them to the closest matching timestamps in the Knowledge Graph. For example, a query for "12:00 to 12:30" would approximate and retrieve the relevant data, even if the exact timestamps stored in the Knowledge Graph are "12:06:12 to 12:47:57." This improvement would significantly increase the flexibility and user-friendliness of the querying process, enabling users to ask more natural and practical questions.

Listing 3: SPARQL Query to retrieve the motor status of *machine_2*

```
PREFIX smo: <http://www.semanticweb.org/manufacturingproductionline/>
PREFIX d: <http://www.semanticweb.org/manufacturingproductionline/data/>
PREFIX tm: <http://www.w3.org/2006/time#>
SELECT DISTINCT ?Motor_Name ?Status ?Start_time
WHERE { d:Machine_1 smo:hasTool ?motor.
?motor smo:hasName ?Motor_Name.
?motor smo:hasMotorState ?state.
?process tm:hasTime ?time.
?state smo:hasState ?Status.
?time tm:hasStartTime > "2021-06-01T 10:11:00Z"^^xsd:dateTime &&
?Start_time < "2021-06-01T 10:12:55Z"^^xsd:dateTime).}</pre>
```

5. Conclusion and Future Work

This study investigates the application of LLMs to improve the usability and interpretability of KGs. Through the utilisation of FAISS-based sentence-transformer to turn our KG into a vector store and the integration of a Python-adapted C++-compiled version of LLaMA2 from Meta AI. We have successfully showcased the ability of prompt engineering to produce simple and contextually appropriate explanations for retrieved triples in response to user enquiries. This method, which bypasses the resource-intensive process of comprehensive model training, greatly enhances the accessibility and understanding of data. Our future research will prioritise advanced prompt engineering, enhancing the KG by including more data sources, optimising system efficiency, integrating user feedback, and investigating cross-domain ontology mapping to develop a more resilient and adaptable.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number SFI/12/RC/2289_P2 (Insight). For Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

- [1] Yahya, Muhammad, Abdul Wahid, Lan Yang, John G. Breslin, Evgeny Kharlamov, and Muhammad Intizar Ali. "Link Prediction in Industrial Knowledge Graphs: A Case Study on Football Manufacturing." IEEE Access (2024).
- [2] Hogan, Aidan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane et al. "Knowledge graphs." ACM Computing Surveys (Csur) 54, no. 4 (2021): 1-37.
- [3] Zhou, Baifan, Zhipeng Tan, Zhuoxun Zheng, Dongzhuoran Zhou, Yunjie He, Yuqicheng Zhu, Muhammad Yahya et al. "Neuro-Symbolic AI at Bosch: Data Foundation, Insights, and Deployment." In ISWC (Posters/Demos/Industry). 2022.
- [4] Sabou, Marta, Fajar J. Ekaputra, Tudor Ionescu, Juergen Musil, Daniel Schall, Kevin Haller, Armin Friedl, and Stefan Biffl. "Exploring enterprise knowledge graphs: A use case in software engineering." In The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, pp. 560-575. Springer International Publishing, 2018.
- [5] Atemezing, Ghislain Auguste. "Applying Semantic Web Technologies to Assess Maintenance Tasks from Operational Interruptions: A Use-Case at Airbus." In The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part II 14, pp. 3-17. Springer International Publishing, 2017.
- [6] Yahya, Muhammad, Aedan Breathnach, Faisal Khan, Iman Abaspur, and Rajkumar Ranganathan. "Towards Semantic Modeling of Camera from Image Quality Testing Perspective: Valeo Vision Systems Case." In SemIIM. 2023.
- [7] Khayatbashi, Shahrzad, Sebastián Ferrada, and Olaf Hartig. "Converting property graphs to RDF: a preliminary study of the practical impact of different mappings." In Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), pp. 1-9. 2022.
- [8] Pan, Shirui, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying large language models and knowledge graphs: A roadmap." IEEE Transactions on Knowledge and Data Engineering (2024).
- [9] Hadi, Muhammad Usman, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, and Seyedali Mirjalili. "A survey on large language models: Applications, challenges, limitations, and practical usage." Authorea Preprints (2023).
- [10] Banerjee, Agniva, Raka Dalal, Sudip Mittal, and Karuna Pande Joshi. "Generating digital twin mod-

els using knowledge graphs for industrial production lines." In Workshop on Industrial Knowledge Graphs, co-located with the 9th International ACM Web Science Conference 2017. 2017.

- [11] Stavropoulou, Georgia, Konstantinos Tsitseklis, Lydia Mavraidi, Kuo-I. Chang, Anastasios Zafeiropoulos, Vasileios Karyotis, and Symeon Papavassiliou. "Digital Twin Meets Knowledge Graph for Intelligent Manufacturing Processes." Sensors 24, no. 8 (2024): 2618.
- [12] Yahya, Muhammad, John G. Breslin, and Muhammad Intizar Ali. "Semantic web and knowledge graphs for industry 4.0." Applied Sciences 11, no. 11 (2021): 5110.
- [13] Ali, Waqas, Muhammad Saleem, Bin Yao, Aidan Hogan, and Axel-Cyrille Ngonga Ngomo. "A survey of RDF stores & SPARQL engines for querying knowledge graphs." The VLDB Journal (2022): 1-26.
- [14] Xiao, Youzi, Shuai Zheng, Jiancheng Shi, Xiaodong Du, and Jun Hong. "Knowledge graph-based manufacturing process planning: A state-of-the-art review." Journal of Manufacturing Systems 70 (2023): 417-435.
- [15] Grangel-González, Irlán, Felix Lösch, and Anees ul Mehdi. "Knowledge graphs for efficient integration and access of manufacturing data." In 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, pp. 93-100. IEEE, 2020.
- [16] Tan, Zhipeng, Baifan Zhou, Zhuoxun Zheng, Ognjen Savkovic, Ziqi Huang, Irlan-Grangel Gonzalez, Ahmet Soylu, and Evgeny Kharlamov. "Literal-aware knowledge graph embedding for welding quality monitoring: a bosch case." In International Semantic Web Conference, pp. 453-471. Cham: Springer Nature Switzerland, 2023.
- [17] Li, Yihao, Ru Zhang, Jianyi Liu, and Gongshen Liu. "An enhanced prompt-based LLM reasoning scheme via knowledge graph-integrated collaboration." arXiv preprint arXiv:2402.04978 (2024).
- [18] Li, Harry, Gabriel Appleby, and Ashley Suh. "A Preliminary Roadmap for LLMs as Assistants in Exploring, Analyzing, and Visualizing Knowledge Graphs." arXiv preprint arXiv:2404.01425 (2024).
- [19] Zhang, Xin, Tianjie Ju, Huijia Liang, Ying Fu, and Qin Zhang. "LLMs Instruct LLMs: An Extraction and Editing Method." arXiv preprint arXiv:2403.15736 (2024).
- [20] Zhu, Yuqi, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. "Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities." arXiv preprint arXiv:2305.13168 (2023).
- [21] Meyer, Lars-Peter, Claus Stadler, Johannes Frey, Norman Radtke, Kurt Junghanns, Roy Meissner, Gordian Dziwis, Kirill Bulert, and Michael Martin. "Llm-assisted knowledge graph engineering: Experiments with chatgpt." In Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow, pp. 103-115. Wiesbaden: Springer Fachmedien Wiesbaden, 2023.
- [22] Kommineni, Vamsi Krishna, Birgitta König-Ries, and Sheeba Samuel. "From human experts to machines: An LLM supported approach to ontology and knowledge graph construction." arXiv preprint arXiv:2403.08345 (2024).
- [23] Shu, Dong, Tianle Chen, Mingyu Jin, Yiting Zhang, Mengnan Du, and Yongfeng Zhang. "Knowledge Graph Large Language Model (KG-LLM) for Link Prediction." arXiv preprint arXiv:2403.07311 (2024).
- [24] Sun, Kai, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. "Head-to-tail: How knowledgeable are large language models (llm)? AKA will llms replace knowledge graphs?." arXiv preprint arXiv:2308.10168 (2023).
- [25] Yahya, Muhammad, Aabid Ali, Qaiser Mehmood, Lan Yang, John G. Breslin, and Muhammad Intizar Ali. "A benchmark dataset with Knowledge Graph generation for Industry 4.0 production lines." Semantic Web 15, no. 2 (2024): 461-479.
- [26] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bertnetworks." arXiv preprint arXiv:1908.10084 (2019).