

LLM-Driven Knowledge Graphs: Automated Creation and Natural Language Querying for Materials Science Research

Vladislav Kaverinskiy^{1,*}, Oeksandr Palagin^{1,*}, Dariia Nikitiuk¹, Anna Litvin¹,
Kyrylo Malakhov¹

¹ Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Glushkov av. 40 03187 Kyiv, Ukraine

Abstract

The presented work is devoted to integrating large language models (LLMs) into knowledge graph construction and query generation presents a transformative opportunity in scientific domains like materials science. Namely, this study explores using LLMs – specifically GPT-4, DeepSeek, and Qwen2.5-72B-Instruct – to automate the creation of knowledge graphs from scientific articles and generate Cypher queries from natural language inputs. A multi-step methodology was developed, involving JSON extraction, RDF/XML conversion, and deployment into graph databases, with iterative meta-learning to refine query accuracy. Four knowledge graph variants were evaluated, with the "Qwen-GPT-4" combination emerging as the most comprehensive due to its detailed entity linkages and structural coherence. Results demonstrate that iterative LLM prompting significantly enhances Cypher query generation and addresses issues such as mislabeled relationships and parasitic nodes. This work highlights the potential of LLMs to streamline knowledge management, aligning and enhancing accessibility to complex scientific data.

Keywords

natural language processing, large language models, prompt engineering, knowledge graph, knowledge management

1. Introduction

In materials science, the growing volume of research data requires efficient methods to organise and interrogate knowledge. Traditional approaches to constructing knowledge graphs – manually curating entities and relationships – are labour-intensive and prone to scalability issues. Recent advances in large language models (LLMs), however, offer a promising alternative: automating the extraction of structured information from unstructured texts and enabling natural language interfaces for querying graph databases.

This study addresses two interconnected challenges: automating knowledge graph creation from scientific articles using LLMs and generating accurate Cypher queries from natural language inputs through iterative model refinement. By means of models such as GPT-4, DeepSeek, and Qwen2.5-72B-Instruct, a pipeline was developed that transforms articles into RDF/XML-based graphs deployable in systems like Neo4J. Furthermore, it has been introduced a meta-learning

Workshop "Software engineering and semantic technologies" SEST, co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13-14, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ insamhlaithe@gmail.com (V.Kaverinskiy); palagin_a@ukr.net (O.Palagin); nikityuk.dv@gmail.com (D.Nikitiuk); litvin_any@ukr.net (A.Litvin); k.malakhov@incyb.kiev.ua (K.Malakhov)

ORCID 0000-0002-6940-579X (V.Kaverinskiy); 0000-0003-3223-1391 (O.Palagin); 0000-0002-9639-5232 (D.Nikitiuk); 0000-0002-5648-9074 (A.Litvin); 0000-0003-3223-9844 (K.Malakhov)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

approach where LLMs iteratively improve Cypher query generation by learning from corrected outputs.

This work builds on prior research in LLM-driven SPARQL/Cypher generation and ontology extraction, yet extends these efforts by focusing on domain-specific challenges in materials science, such as capturing process-structure-property relationships. The results underscore the viability of LLMs in democratising access to complex datasets while highlighting critical considerations for model selection, prompt engineering, and structural validation. By integrating iterative meta-learning, we address challenges such as parasitic nodes and mislabelled relationships – issues akin to those observed in multilingual text processing systems [1], where ontology-driven approaches improve cross-lingual consistency.

The integration of LLMs into knowledge graph construction and query generation represents a paradigm shift in managing scientific data, particularly in domains like materials science. While traditional approaches to knowledge extraction rely on manual curation or rigid rule-based systems [2], recent advancements in semantic analysis and model-driven engineering offer pathways to automation. For instance, [3, 4] demonstrated the viability of semantic matching for software asset reuse, leveraging keywords and OCL expressions to align requirements with existing components – a methodology that parallels the semantic grounding of entities in knowledge graphs. Similarly, multilingual frameworks for text-to-model transformations [1] highlight the potential of structured representations (e.g., UML diagrams) to bridge unstructured texts and formalized knowledge structures, a concept critical to graph-based data interoperability.

The primary objective of this study is to develop and validate a framework that utilizes large language models to automate the construction of domain-specific knowledge graphs from unstructured scientific articles and enable intuitive querying through natural language interfaces. By integrating iterative meta-learning, the approach refines Cypher query generation for graph databases, addressing challenges in entity linkage accuracy and structural coherence. Focused on materials science, this work aims to enhance data accessibility and reduce manual curation efforts, thereby streamlining knowledge management in research workflows.

2. Related Works

The integration of LLMs into the generation of structured data representations, such as RDF/XML knowledge graphs and SPARQL queries, has emerged as a transformative research area. Recent advancements highlight the potential of LLMs to bridge the gap between natural language inputs and formalized knowledge structures, enabling more accessible interaction with complex datasets. Central to this progress is the ability of LLMs to translate user queries into structured query languages like SPARQL and Cypher, which are critical for interacting with graph databases and knowledge graphs. For instance, Emonet et al. [5] introduced a Retrieval-Augmented Generation (RAG) system that leverages LLMs alongside metadata, including query examples and schema information, to generate federated SPARQL queries over bioinformatics knowledge graphs. Their validation step reduces hallucinations, improving reliability. Similarly, Mecharnia and d’Aquin [6] demonstrated the effectiveness of fine-tuned LLMs in converting natural language questions into SPARQL queries, though they noted challenges in handling domain-specific nuances and logical consistency. In the realm of Cypher, Ozsoy et al. [7] developed Text2Cypher, a system that translates natural language into graph database queries, emphasizing the role of high-quality datasets and fine-tuning in improving performance. These efforts underscore the dual focus on enhancing user accessibility and ensuring query accuracy, particularly in scientific domains such as materials science.

Materials science has seen significant advancements through the application of ontologies and graph databases. Dreger et al. [8] proposed a native graph database architecture to store heterogeneous data from fabrication workflows, measurements, and simulations, extending the European Materials Modelling Ontology (EMMO) to standardize energy materials data. This approach aligns with the broader goals of FAIR (Findable, Accessible, Interoperable, Reusable) data

principles. A complementary study introduced a materials graph ontology in [9], addressing gaps in existing ontologies by formalizing data ingest frameworks to capture process-structure-property relationships. These developments highlight how LLMs can further enhance such systems by automating ontology creation and query generation, though challenges remain in ensuring logical validity and scalability.

The evolution of neural machine translation (NMT) architectures for SPARQL generation has been pivotal. Yin et al. [10] compared CNNs, RNNs, and Transformers, finding that CNN-based models achieve high BLEU scores and accuracies on datasets like Monument and DBNQA. However, the structured nature of SPARQL and out-of-vocabulary (OOV) issues persist as challenges. Hirigoyen et al. [11] addressed OOV by integrating a copy mechanism into encoder-decoder architectures, enabling the direct transfer of knowledge base tokens from input questions to queries. This approach mitigates ambiguities in schema elements, a critical step toward robust query generation. Parallel advancements in visiolinguistic learning, such as survey [12], emphasize the role of external knowledge graphs and LLMs in tasks like Visual Question Answering (VQA) and Image Captioning. These studies advocate hybrid models combining explicit knowledge (e.g., ontologies) with implicit knowledge (e.g., pre-trained models) to enhance multimodal reasoning.

PAROT, a dependency-based framework for SPARQL generation [13], exemplifies the synergy between syntactic analysis and ontology alignment. Its lexicon, built using the lemon model, resolves ambiguities in scalar adjectives and negation, while dependency parsing identifies triples and logical operators. Evaluated on QALD-9 and Geoquery datasets, PAROT outperformed gAnswer in complex queries, achieving 87.55% Macro-F1 on Geoquery. However, its reliance on dependency parsing introduces computational overhead, and temporal query handling remains limited. These limitations underscore the need for integrative approaches, such as combining LLMs with symbolic reasoning, to address scalability and domain-specific requirements.

The automation of OWL ontology creation from scientific texts represents a frontier in LLM applications. Current systems face challenges in ensuring logical consistency and semantic coherence, as OWL requires strict adherence to description logic. Recent studies, such as the work by Abolhasani and Pan [14], explore LLMs for ontology extraction, leveraging structured prompts and iterative refinement. For instance, OntoKGen employs a Chain of Thought algorithm to align outputs with user requirements, while fusion-jena’s semi-automated pipeline constructs knowledge graphs from competency questions. Despite these advances, accuracy and validation remain critical issues, particularly in specialized domains like materials science.

Comparative analyses of LLMs reveal distinct strengths and limitations. GPT-4 demonstrates broad code generation capabilities but requires post-validation for OWL axioms. Qwen-72B excels in API-centric tasks and multilingual contexts but struggles with low-resource programming languages. DeepSeek-MoE, optimized for STEM domains, incorporates lightweight reasoning to reduce inconsistencies, as seen in its biomedical ontology experiments [15]. These models highlight the importance of domain-specific fine-tuning and hybrid architectures that blend LLM flexibility with symbolic constraints.

The evaluation of LLM-based systems often relies on benchmarks like HumanEval and HumanEval-Math, though metrics for ontology quality remain underdeveloped. Functional correctness, measured via pass@k rates, is supplemented by structural metrics like CodeBLEU, which assesses syntax trees. However, assessing logical validity in OWL requires specialized tools, such as reasoners to detect entailment violations. Future research must prioritize standardized benchmarks and interdisciplinary methods that merge knowledge representation with deep learning.

Challenges persist in ambiguity resolution, scalability, and ethical considerations. Natural language descriptions often lack precision, leading to inconsistent axioms or queries. Temporal reasoning and aggregation operations remain underexplored, limiting applications in dynamic datasets. Additionally, biases in training data and knowledge representation can propagate errors, necessitating rigorous validation frameworks. The integration of LLMs with existing tools like

Protégé and graph databases offers a path forward, enabling iterative refinement and human oversight.

In materials science, the synergy between LLMs and ontologies has practical implications for data interoperability. For example, Dreger et al. [8] extended EMMO to capture fabrication workflows, enabling systematic analysis of material properties. LLMs could automate this process by extracting entities and relationships from research articles, though domain-specific training and prompt engineering are essential. Experiments demonstrated varying success across GPT, Qwen, and DeepSeek in generating ontologies from scientific texts, with DeepSeek's biomedical focus yielding the most consistent results.

The future of LLM-driven knowledge representation lies in hybrid architectures, dataset enrichment, and interdisciplinary collaboration. Integrating LLMs with symbolic reasoning systems, such as those proposed by Nakajima and Miura [16], could enhance logical rigour. Expanding training data with domain-specific annotations and leveraging federated learning for sensitive datasets are additional strategies. Ethical frameworks must also evolve to address transparency and accountability, ensuring that automated systems align with scientific and societal values.

Thus, LLMs have revolutionized the translation of natural language into structured data formats, offering unprecedented opportunities for scientific research and data management. While challenges in consistency, scalability, and domain adaptation remain, ongoing advancements in model architectures, evaluation methodologies, and hybrid systems promise to unlock the full potential of these tools. Prior research in semantic analysis and model-driven engineering provides foundational insights for LLM-driven knowledge graph workflows. In [1] it was introduced semantic matching techniques for software reuse, using OCL expressions to compare requirement specifications with repository assets – a precursor to LLM-based entity linkage in knowledge graphs. Meanwhile, text-to-model transformation frameworks [17] underscore the importance of structured representations, such as XMI and PlantUML, in restoring UML diagrams from heterogeneous formats, a challenge mirrored in RDF/XML graph conversions.

3. Materials and Methods

The presented here study included two main parts. The first is the creation of a knowledge graph grounded on a set of scientific articles involving LLMs. The second part was devoted to the creation of Cypher queries to the developed graph database from natural language queries using an LLM. The input articles [18 – 22] all belong to the domain of material science. The following LLMs have been use in the study: GPT-4 [23], Deep Seek (R1) [24], and Qwen2.5-72B-Instruct [25].

For the creation of the knowledge graph the following approach has been developed, which includes the next main steps:

1. Develop a prompt for LLM that includes the list and descriptions of the desired classes of the graph nodes and links between them. The prompt also includes a template of the JSON structure to be used for the extracted information storage.
2. Creation of a JSON representation of the knowledge graph using an LLM with the prompt and input articles. At this stage, several articles at once could be used, if the context window of the LLM allows it or the JSON representations are created by one for each of the articles.
3. A merged knowledge graph in JSON format is created from the obtained on the previous step fragments by means of an LLM. For the knowledge graph merge either the same or different LLM can be used, depending on the quality of the obtained result.
4. The merged knowledge graph from the JSON format also using an LLM is to be transformed to RDF/XML format, which is needed to easily export it to a graph DBMS like Neo4J [26] or Apache Jena Fuseki [27].
5. The final step is deploying the knowledge graph into a chosen graph DBMS.

The scheme of this technique is presented in Figure 1.

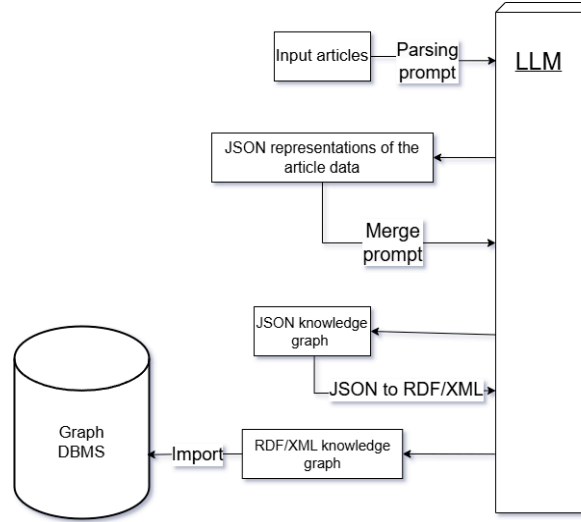


Figure 1: A general scheme of the involved knowledge graph from articles creation technique.

The text of the prompt was the same for all the considered LLMs. The text of the prompt as well as the obtained JSON and RDF/XML representations of the knowledge graphs can be found in the GIT Hub repository [28].

For the interface of Qwen allows only a single file downloading in one message the JSON representations were being created for each of the given articles and then merged into a one file using an LLM. Other ones – GPT and Deep Seek – allow several files uploading, however still have limits for the attached files total size.

During the carried out study, the following four variants of RDF/XML knowledge graphs have been created:

1. Documents data parsing in Deep Seek, conversion to RDF/XML using GPT-4 (“DS – GPT-4”).
2. Documents data parsing in Deep Seek, conversion to RDF/XML using Qwen (“DS – Qwen”).
3. Documents data parsing in Qwen, merging and conversion to RDF/XML using GPT-4 (“Qwen – GPT-4”).
4. Documents data parsing in Qwen, conversion to RDF/XML using Qwen – one file without merging (“Qwen – Qwen”).

For experiments with Cypher queries a knowledge graph “Qwen – GPT-4” has been selected as containing information from all the articles as well being found rather more complete and well structured than others.

The following technique has been applied for Cypher query creation from natural language ones using LLM (GTP-4). An initial prompt has been provided:

```

Here is an RDF/XML file. This file can be imported to Neo4J. You
following tasks will be generation of Cypher queries from natural
nanguage queries. The init parameters for import were:
handleVocabUris: "IGNORE", handleMultival: "OVERWRITE",
handleRDFTypes: "LABELS".
  
```

As it can be seen, an appropriate RDF/XML has been given to LLM to inform it with the knowledge graph structure. Also, some additional technical information has been provided in this prompt, namely the graph import parameters. Then the LLM was been asked to create a certain query for a rather simple natural language phrase. The returned Cypher query execution was

tested in Neo4J. Then the text of the query was manually changed to improve the output. This revised version of the query was being provided for LLM for further generation improvement. Then the LLM has been provided with new natural language queries – more complicated and/or devoted to different subjects. The reason for the procedure was iterative interactive meta-learning of an LLM to improve its ability for Cypher query generation tuned for a certain knowledge base.

4. Results and Discussion

All of the four created RDF/XML representations of the knowledge graphs appear valid – able to be opened in Protégé editor and have been successfully imported to Neo4J. However, “DS – GPT-4” have demonstrated quite a specific structure, so the entity names did not appear as convenient classes and properties. The fullness of each of them can be judged from the RDF triplets numbers counted when knowledge graphs are imported to Neo4J. The corresponding values are presented in Table 1.

Table 1

Estimated sizes the generated knowledge graph

Knowledge graph (by LLMs used)	Number or RDF triples	File size, Kb
DS – GPT-4	129	12.4
DS – Qwen	126	14.1
Qwen – GPT-4	207	23.8
Qwen – Qwen	201	20.6

The knowledge graph “DS – GPT-4” has appeared rather brief and mostly thesaurus-like. Each entity description contains tags <rdfs:label> and <rdfs:comment> to provide their appropriate name and description. However, the descriptions appeared quite brief and not very informative. Instead of the usual <rdf:type> tag this representation includes an <ex:category> container tag. In general this knowledge graph includes not very developed linkage between entities and looks rather as a demonstrative example.

The next “DS – Qwen” knowledge graph has some similarities to the previously mentioned one, which is not surprising for Deep Seek LLM was used in the main operation of knowledge extraction for both of them. The descriptions are still brief but now are included in tags <ex:description> and instead of <rdfs:label> here we found <ex:name>. However, the class belonging identifiers now are in <rdf:type> tags. So when parsing it through Protégé or Neo4J the types (classes) of the nodes explicitly appeared and were accessible. However, as well as previously, here we see quite poor rather demonstrative linkage. It seems like Deep Seek LLM works in a “lazy” way, especially when dealing with a set of files. It extracts some random linked data and then allows you to proceed in the same way behaving rather like a virtual assistant than a working tool which it is expected to be regarded to the current task.

Somewhat better results have been obtained by processing the input files one by one using Qwen 72B-Instruct for the purpose. Having less input information but operating with the same amount of the context window volume more information could be extracted from each of the input documents. An example of such an approach here is the “Qwen – GPT-4” knowledge graph. It has a rather more developed linkage and covers most of the key aspects presented in the articles. In this version of our knowledge graph, the most detailed and comprehensive descriptions are provided. Moreover, some of such descriptions contain not only presented in the input article information but also details collected elsewhere, probably owing to the web search possibility of the LLMs used.

Dealing with the stage of RDF/XML formation with only one document JSON representation. It does not contain such detailed and comprehensive descriptions, but the biggest number of entities

related to one article. One of the peculiarities of this knowledge graph is named intermediate resource nodes with <ex:has_value> tags that represent the qualitative values of the links. The links themselves do not include values but directions only. The “Qwen – GPT-4” knowledge graph either does not have values of the links but has “parasitic” nodes built between the reasonable nodes. Such “parasitic” nodes we can see as well in the structure of the “DS – GPT-4” knowledge graph and their presence could be caused by some drawbacks brought in by some GPT-4 behaviour hangs on the RDF/XML creation stage. It seems like Qwen represents the link value explicitly but GPT-4 is not.

The “Qwen – GPT-4” knowledge graph has been selected for further study of Cypher queries creation for several main reasons: it covers several articles of different topics but some similarities; it is rather developed in linkage than “DS – GPT-4” and “DS – Qwen”; it has those “parasitic” nodes which are an interesting challenge for LLM to overcome when queries creation meta-learning; it has comprehensive descriptions which seam perspective for further practical implementation.

Let us consider an example of meta-learning of Cypher query creation using GPT-4o with reasoning. The first task will be: “Give me the names of the article and the corresponding keywords.”. The first resulting Cypher query returned by the LLM was as follows:

```
MATCH (a:Article)-[:HAS_KEYWORD]->(kw:Keyword)
RETURN a.name AS articleName, kw.name AS keyword
```

This query is generally syntactically correct, but does not assume all the specifics regarding to a certain knowledge graph. Namely, the correct link name here must be “includes_terms” not “HAS_KEYWORD”, the class of the node is “Key_word” but not “Keyword”. The class “Article” is correct, but here is not obligatory, because in the current graph all the “Key_word” have incoming links from “Articles” only. Thereby, more correct query will be the next:

```
MATCH (n)-[:includes_terms]->(m:Key_word)
RETURN n.name AS articleName, m.name AS keyword
```

The following task was to create a Cypher query from: “Give me the names of the articles by the keyword “Steel”. This time the LLM coped with the job and returned the next correct query:

```
MATCH (n)-[:includes_terms]->(m:Key_word {name:"Steel"})
RETURN n.name AS articleName
```

The query have returned a correct result of two articles names (“Mathematical Modelling of Primary Recrystallization Kinetics and Precipitation of Carbonitride Particles in Steels. II. Recrystallization Kinetics” and “Mathematical Modelling of Kinetics of Decomposition of the Supercooled Austenite in Alloyed Steels”) which indeed include the keyword “Modelling” in the knowledge graph.

Then a different and a bit more complicated task was given to LLM, namely to create a Cypher query for the request: “Provide the names and descriptions of the topics to which the articles belong.” The first returned result was again not correct:

```
MATCH (a)-[:belongs_to]->(t:Topic)
RETURN DISTINCT t.name AS topicName, t.description AS
topicDescription
```

The two main reasons are not assuming a “parasitic” node between “Article” and “Topic”, and, a minor one, the real name of the link is “belongs_to_topic” not just “belongs_to”. Thus, the correct version will be:

```
MATCH (n)-[:belongs_to_topic]-(m)-[:links_to]-(c:Topic)
RETURN n.name as Article, c.name AS Topic, c.description AS
Description
```

Here appears that “parasitic” node (m) with its following link “links_to”. Also a not critical but making the response more convenient will be returning the name of the article as well.

The following query to the LLM to make a Cypher query was: “To which topic belong articles that have the keyword “Phase Diagrams”?”. The response of LLM this time was almost completely correct:

```

MATCH      (n)-[:includes_terms]->(kw:Key_word      {name:"Phase
Diagrams"}),
      (n)-[:belongs_to_topic]-(m)-[:links_to]-(c:Topic)
RETURN DISTINCT c.name AS Topic, c.description AS Description

```

To improve this query the following prompt has been provided to LLM:

This is almost correct. A correct result has been returned. But there was not a request for the description, but instead it would be better to include also the name of the article to the results.

After this notice the next corrected version of the query was returned:

```

MATCH      (n)-[:includes_terms]->(kw:Key_word      {name:"Phase
Diagrams"}),
      (n)-[:belongs_to_topic]-(m)-[:links_to]-(c:Topic)
RETURN n.name AS Article, c.name AS Topic

```

This one returns articles matched to their topics, but without descriptions, which it was not asked for.

For the request to make a Cypher query for the natural language query: “Give me the name and description of the technology devoted to the phase transformation process”. A completely correct response has been obtained, which is grounded on the previous tries and fails:

```

MATCH (a:Article)-[]-(r)-[:links_to]-(n:Technology)-
[:include_process]-(m)-[:links_to]-
(c:Process {name:"Phase Transformation"})
RETURN n.name AS Technology, n.description AS Description, a.name as InArticle

```

Such an interactive iterative process of meta-learning of an LLM seems to be a promising approach to adopt such a model to a certain knowledge graph to convert natural language requests to formal Cypher queries. This could be useful for natural language reference systems development which use in their structure APIs of large language models as a working tool.

In addition, the semantic coherence observed in our Cypher queries aligns with findings from software reuse experiments [3, 4], where AI-driven comparisons of user stories reduced development time. However, challenges persist in entity labelling consistency, akin to limitations in multilingual systems handling natural languages of different types [1, 17]. Future work could integrate domain-specific fine-tuning, as proposed for PlantUML-based UML generation [17, 29], to improve contextual awareness. By addressing these challenges, our framework not only streamlines knowledge management but also advances compliance with FAIR principles [30], echoing the transformative potential of LLMs in software engineering and cross-lingual data interoperability.

5. Conclusions

This study demonstrates the efficacy of LLMs in automating knowledge graph construction and query generation for materials science research. By comparing four LLM-generated graph variants, the "Qwen-GPT-4" combination proved superior, balancing structural coherence with comprehensive entity linkages. The iterative meta-learning approach for Cypher query generation – where models adapt to correct relationship labels and navigate parasitic nodes – significantly enhanced accuracy, enabling robust natural language interfaces.

Challenges persist, particularly in ensuring consistent entity labelling and minimising extraneous nodes introduced during RDF conversion. Future work could explore hybrid architectures combining LLMs with symbolic reasoning tools to enforce logical rigour, or domain-specific fine-tuning to improve contextual awareness. Additionally, expanding benchmarking metrics to assess ontological validity and scalability remains critical.

Ultimately, this approach offers a pathway to streamline knowledge management in scientific domains. By reducing reliance on manual curation, LLMs can accelerate research workflows, foster interoperability, and make complex datasets more accessible to both specialists and non-experts. As LLM capabilities evolve, integrating them with knowledge graphs promises to unlock new frontiers in data-driven research.

Acknowledgements

This research was carried out as part of the scientific and technical project “Develop means of supporting virtualization technologies and their use in computer engineering and other applications” (state registration number: 0124U001826). The study was also conducted under the scientific and technical project “To develop theoretical foundations and a functional model of a computer for processing complex information structures” (state registration number: 0124U002317). Both projects are being implemented at the V. M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Kyiv, Ukraine.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

CRedit authorship contribution statement

Vladislav Kaverinsky: Investigation, Writing – Original Draft, Resources, Methodology. Oleksandr Palagin: Conceptualization, Supervision. Dariia Nikitiuk: Writing – Review & Editing. Anna Litvin: Writing – Review & Editing. Kyrylo Malakhov: Validation, Writing – Review & Editing.

References

- [1] O. Chebanyuk, Multilingual Question-Driven Approach and Software System to Obtaining Information From Texts, in: CEUR Workshop Proceedings, vol. 3501, 2022, pp. 256–265. URL: <https://ceur-ws.org/Vol-3501/s24.pdf>.
- [2] A. Litvin, O.V. Palagin, V.V. Kaverinskiy, K.S. Malakhov, Ontology-Driven Development of Dialogue Systems, South African Computer Journal 35 (1), 37–62 (2023). doi:10.18489/sacj.v35i1.1233.
- [3] O. Chebanyuk, An Approach to Software Assets Reusing, in: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, vol. 450 of LNICST, 2022, pp. 73–83. doi:10.1007/978-3-031-17292-2_6.
- [4] O. Chebanyuk, Requirement Analysis Approach to Estimate the Possibility of Software Development Artifacts Reusing Consulting with Artificial Intelligence Technologies, in: CEUR Workshop Proceedings, vol. 3806, 2024, pp. 62–74. URL: https://ceur-ws.org/Vol-3806/S_43_Chebanyuk.pdf.
- [5] V. Emonet, J. Bolleman, S. Duvaud, T. M. de Farias, A. C. Sima, LLM-based SPARQL Query Generation from Natural Language over Federated Knowledge Graphs, arXiv preprint arXiv:2410.06062 (2024).
- [6] T. Mecharnia, M. d’Aquin, Performance and Limitations of Fine-Tuned LLMs in SPARQL Query Generation, in: Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK), 2025, pp. 69–77.
- [7] M. G. Ozsoy, L. Messallem, J. Besga, G. Minneci, Text2Cypher: Bridging Natural Language and Graph Databases, in: Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK), 2025, pp. 100–108.
- [8] M. Dreger, K. Malek, M. J. Eslamibidgoli, M. H. Eikerling, Synergizing Ontologies and Graph Databases for Highly Flexible Materials-to-Device Workflow Representations, J. Mater. Inform. 3 (2023).

- [9] S. P. Voigt, S. R. Kalidindi, Materials Graph Ontology, *Mater. Lett.* 295, 129836 (2021). doi:10.1016/j.matlet.2021.129836.
- [10] X. Yin, D. Gromann, S. Rudolph, Neural Machine Translating from Natural Language to SPARQL, *Future Gener. Comput. Syst.* 117, 510–519 (2021). doi:10.1016/j.future.2020.12.013.
- [11] R. Hirigoyen, A. Zouaq, S. Rey, A Copy Mechanism for Handling Knowledge Base Elements in SPARQL Neural Machine Translation, in: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2022*, 2022, pp. 226–236.
- [12] M. Lymperaio, G. Stamou, The Contribution of Knowledge in Visiolinguistic Learning: A Survey on Tasks and Challenges, *CEUR Workshop Proceedings*, 2023.
- [13] P. Ochieng, PAROT: Translating Natural Language to SPARQL, *Expert Syst. Appl.* 10, 100024 (2020). doi:10.1016/j.eswax.2020.100024.
- [14] M. S. Abolhasani, R. Pan, Leveraging LLM for Automated Ontology Extraction and Knowledge Graph Generation, *arXiv preprint arXiv:2412.00608* (2024).
- [15] L. Zhang et al., Biomedical Ontology Generation with DeepSeek, *J. Biomed. Semantics* (2024). doi:10.1186/s13326-024-00311-4.
- [16] H. Nakajima, J. Miura, Combining LLMs and Symbolic Reasoning for OWL Ontology Generation, in: *Proc. AAAI Conf. Artif. Intell.*, 2024. arXiv:2410.16804v1.
- [17] An Approach of Text to Model Transformation of Software Models, in: *ENASE 2018: Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering*, Funchal, Madeira, Portugal, 2018, pp. 432–439. doi:10.5220/0006804504320439.
- [18] V.V. Kaverynsky, A.I. Trotsan, Z.P. Sukhenko, Mathematical Modelling of Kinetics of Decomposition of the Supercooled Austenite in Alloyed Steels, *Metallofiz. Noveishie Tekhnol.* 39(8), 1051–1068 (2017). doi:10.15407/mfint.39.08.1051.
- [19] V.V. Kaverinsky, Z.P. Sukhenko, Mathematical Modelling of Primary Recrystallization Kinetics and Precipitation of Carbonitride Particles in Steels. I. Precipitation, *Metallofiz. Noveishie Tekhnol.* 43(1), 27–45 (2021). doi:10.15407/mfint.43.01.0027.
- [20] V.V. Kaverinsky, Z.P. Sukhenko, Mathematical Modelling of Primary Recrystallization Kinetics and Precipitation of Carbonitride Particles in Steels. II. Recrystallization Kinetics, *Metallofiz. Noveishie Tekhnol.* 43(2), 235–244 (2021). doi:10.15407/mfint.43.02.0235.
- [21] V.V. Kaverinsky, Z.P. Sukhenko, G.A. Bagluk, D.G. Verbylo, About Al–Si Alloys Structure Features and Ductility and Strength Increasing after Deformation Heat Processing, *Metallofiz. Noveishie Tekhnol.* 44(6), 769–784 (2022). doi:10.15407/mfint.44.06.0769.
- [22] V.V. Kaverynsky, Z.P. Sukhenko, Evaluation of Computer Model Results for Thermodynamic and Kinetic Calculation of Phase Transformation in a Middle-Carbon Alloyed Steel, *Metallofiz. Noveishie Tekhnol.* 47(2), 183–197 (2025). doi:10.15407/mfint.47.02.0183.
- [23] OpenAI, GPT-4 Technical Report, *arXiv preprint arXiv:2303.08774* (2023). doi:10.48550/arXiv.2303.08774.
- [24] Alibaba, Qwen Technical Report, 2023. [Online]. URL: <https://qwenlm.github.io>.
- [25] DeepSeek, Lightweight Reasoning in DeepSeek, 2023. [Online]. URL: <https://deepseek.ai/docs/reasoning>.
- [26] Neo4j Inc, Neo4j graph database, 2025. URL: <https://github.com/neo4j/neo4j>
- [27] The Apache Software Foundation, Apache jena kuseki, 2025. [Online]. URL: <https://github.com/apache/jena>.
- [28] GitHub Repository, "Files for the Article Approach to Knowledge Graphs Creatusing Using LLMs," 2024. URL: <https://github.com/VladislavKaverinskiy/Files-for-the-article-Approach-to-knowledge-graphs-creatusing-using-LLMs>.
- [29] O. Chebanyuk, An Approach to Class Diagram Design, in: *Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2014)*, Lisbon, Portugal, 2014, pp. 448–453. doi:10.5220/0004763504480453.
- [30] M.D. Wilkinson et al., The FAIR Guiding Principles for Scientific Data Management and Stewardship, *Sci. Data* 3, 160018 (2016).